TSN 测试仪 3.0

设计方案

（版本 1.0）

OpenTSN 开源项目组

2021 年 5 月

版本历史

版本	修订时间	修订内容	修订人	文件标识
1.0	**2021.5**	初版编制	吴茂文	**TSN 测试仪 3.0**

目录

1、概述

TSN 测试仪是应用于 TSN 网络之间的数据适配节点，主要功能是将应用数据按照流量传输需求以及 TSN 网络封装格式注入到 TSN 网络中进行数据传输，以及将从 TSN 网络接收的数据进行封装返回应用程序并进行相关性能统计。具体包括软件配置能力、IEEE 1588 时钟同步能力、在确定的时间点生成发送时间敏感流、时间感知整形（TAS）、时间敏感流/非时间敏感流的并发、网络流量捕获与分析等。

2、总体设计

2.1. 总体架构

TSN 测试仪硬件逻辑总体架构框图如图 2-1 所示，根据 TSN 测试仪功能需求，对接口应用做了映射（分别用于控制、数据输出、数据输入、流量采样），详见表 2-1。

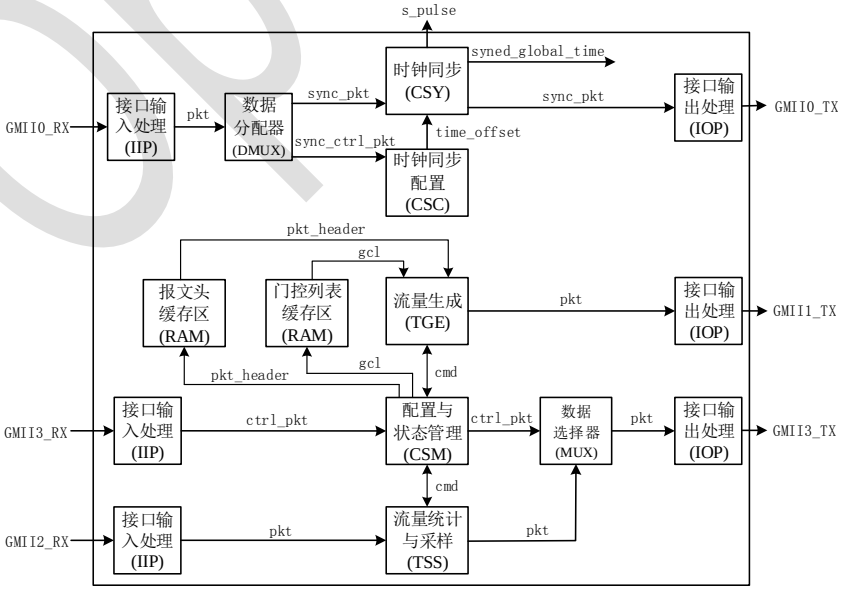

图 2-1 总体架构框图

表 2-1 GMII 接口说明

GMII 接口		说明
GMII0	RX	负责接收 PTP 报文、PTP 封装报文、配置报文（配置时间同步相关参数）。
	TX	负责发送 PTP 报文、PTP 封装报文
GMII1	RX	信号悬空
	TX	负责发送测试仪生成的测试报文
GMII2	RX	负责接收网络中返回的测试报文
	TX	信号输出为固定值 0
GMII3	RX	负责接收配置报文（配置测试报文的特征参数）。
	TX	负责发送采样报文，状态上报报文

IIP（Interface Input Process，接口输入处理）模块：负责丢弃接收报文的前导码、帧开始符和 CRC，将报文传输时钟域从 GMII 接收时钟域切换到到架构内部时钟域，记录时间同步报文接收时间，转换报文位宽。

DMUX（Demultiplexer,数据分配器）模块：负责根据以太网类型对报文进行分派，将时间同步报文、时间同步封装报文分派给时间同步模块，配置报文（配置时间同步相关参数）分派给时钟同步配置模块。

CSY(Clock Synchronization，时间同步)模块：负责记录时间同步报文的接收时间戳、计算透明时钟并更新透明时钟域、将时间同步报文封装到 TSMP（Time Sensitive Management Protocol）帧中，以及将时间同步封装报文解封装为时间同步报文、记录时间同步报文的发送时间戳，以及根据接收到的本地时钟与主时钟的偏差值修正本地时钟。

CSC（Clock Synchronization Configurate，时钟同步配置）模块：负责解析配置报文（配置时钟同步的相关参数），完成对时钟同步相关参数的配置。

CSM(Configuration and State Management，配置与状态管理)模块：负责解析配置报文（配置测试报文的特征参数），完成对本地寄存器和表的配置；以及周期性上报本地的状态信息。

TGE(Traffic Generate，流量生成)模块：基于令牌桶机制、时间感知整形（TAS）来实现多条流的生成与发送，其中用令牌桶机制来控制流量发送速率，用时间感知整形来控制流量发送时刻。

TSS(Traffic Statistic and Sample，流量统计与采样）模块：负责根据五元组匹配（带掩码）统计 TSN 测试仪接收的每条流量的数量以及总接收数量，提取接收报文的五元组并存放在 metadata，然后按一定的采样频率对接收的报文进行封装与采样。

MUX（Multiplexer,数据分配器）模块：负责对 NMAC 状态上报报文和采样的报文进行选择输出。

IOP（Interface Output Process，接口输出处理）模块：负责实现报文位宽转换，增添帧前导符、帧开始符和 CRC，计算时间同步报文在测试仪中传输的透明时钟并更新透明时钟域，将报文传输的时钟域从内部处理时钟域切换到外部 PHY 的时钟域。

2.2 帧的处理流程

2.2.1 时钟同步配置帧解析处理流程

时钟同步配置帧由 GMII0 接口输入，在 GMII3 接口输入处理模块先对帧进行跨时钟域（从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域）转换，接下来将帧每拍数据的位宽由 8bit 转换为 134bit，并增加两拍 metadata，在第一拍 metadata 中携带帧长度和接口接收帧的时间戳，然后将帧输出给数据分配器模块。

数据分配器模块根据帧以太网类型为 0xff01 及其子类型为 0x02 判定该帧为时钟同步配置帧（配置时钟同步参数），然后将时钟同步配置帧分配给时钟同步配置模块。

时钟同步配置模块对接收的时钟同步配置帧进行解析，根据时钟同步配置帧中携带的配置数据个数以及配置基地址，将配置数据写入对应寄存器。

2.2.2 测试参数配置帧解析处理流程

测试参数配置帧由 GMII3 接口输入，在 GMII3 接口输入处理模块先对帧进行跨时钟域（从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域）转换，接下来将帧每拍数据的位宽由 8bit 转换为 134bit，并增加两拍 metadata，在第一拍 metadata 中携带帧长度和接口接收帧的时间戳，然后将帧输出给数据分配器模块。

数据分配器模块根据帧以太网类型为 0xff01 及其子类型为 0x10

或 0x20 判定该帧为测试参数配置帧（配置测试报文的特征参数），然后将测试参数配置帧分配给配置与状态管理模块。

配置与状态管理模块对接收的测试参数配置帧进行解析，若帧以太网类型为 0xff01 及其子类型为 0x10 ，则该测试参数配置帧用以配置 8 种类型流量的报文头，并将该 8 种类型流量的报文头写入缓存区中进行缓存；若帧以太网类型为 0xff01 及其子类型为 0x20 ，则该测试参数配置帧用以配置门控列表、测试报文的长度和发送速率、需统计的流量五元组信息等。在测试参数配置帧配置完成后，配置与状态管理模块给出配置完成信号。

2.2.3 状态上报帧的处理流程

配置与状态管理模块每次接收到上报脉冲后会输出一个上报请求给数据选择器，当接收到数据选择器的响应后开始传输上报报文给数据选择器，上报本地状态（详见表 D-1），包括可配置寄存器、状态寄存器。

数据选择器检测到上报请求时，在没有报文传输时会传给配置与状态管理模块一个响应，接下来开始接收并传输配置与状态管理模块传来的报文给接口输出处理模块。

接口输出处理模块将测试报文的位宽由 134bit 转换为 8bit，以及将报文传输跨时钟域由测试仪内部逻辑工作时钟域切换到 GMII 发送时钟域，然后数据从 GMII3 接口输出。

2.2.4 PTP 帧封装处理流程

PTP 帧由 GMII0 接口输入，在接口输入处理模块先对帧进行跨时钟域（从 GMII 接收时钟域到主时钟内部逻辑工作时钟域）转换，接下来将帧每拍数据的位宽由 8bit 转换为 134bit 后，增加两拍 metadata，在第一拍 metadata 中携带帧长度和接口接收帧的时间戳，然后将帧输出给数据分配器模块。

数据分配器模块根据帧以太网类型为 0x98f7 判定该帧为 PTP 帧，然后分配给时钟同步模块。时钟同步模块将 PTP 报文到达本模块的时间戳减去 PTP 在接口记录的时间戳即得 PTP 报文在测试仪中传输的透明时钟，并将透明时钟累加到透明时钟域中，同时将 PTP 报文到达本模块的时间戳记录在 PTP 报文的 payload 中；将 PTP 报文封装到 TSMP 帧中后传输给接口输出处理模块。

接口输出处理模块将状态上报报文的位宽由 134bit 转换为 8bit，以及将报文传输跨时钟域由测试仪内部逻辑工作时钟域切换到 GMII 发送时钟域，然后数据从 GMII0 接口输出。

2.2.5 PTP 封装帧解封装处理流程

PTP 封装帧由 GMII0 接口输入，在接口输入处理模块先对帧进行跨时钟域（从 GMII 接收时钟域到主时钟内部逻辑工作时钟域）转换，接下来将帧每拍数据的位宽由 8bit 转换为 134bit 后，增加两拍 metadata，在第一拍 metadata 中携带帧长度和接口接收帧的时间戳，然后将帧输出给数据分配器模块。

数据分配器模块根据帧以太网类型为0xff01且子类型为0x5判定该帧为 PTP 封装帧，然后分配给时钟同步模块。时钟同步模块去掉 PTP 封装帧的 TSMP 以太网头，将 PTP 封装帧解封装为 PTP 帧，并且记录 PTP 帧输出本模块的时间戳并将其存放在 payload 中。

接口输出处理模块将 PTP 帧到达本模块的时间戳减去 PTP 帧从时钟同步模块输出的时间戳即得 PTP 报文在测试仪中传输的透明时钟，并将透明时钟累加到透明时钟域中，然后将 PTP 报文的位宽由 134bit 转换为 8bit，以及将报文传输跨时钟域由测试仪内部逻辑工作时钟域切换到 GMII 发送时钟域，然后数据从 GMII0 接口输出。

2.2.6 测试报文发送处理流程

当配置与状态管理模块完成了测试报文的配置且测试仪时钟已和网络时钟同步后，流量生成模块基于令牌桶机制来控制每条流量的发送速率，基于门控列表来控制流量的发送时刻；根据优先级调度策略，来对 8 条流量进行调度，并根据配置的每条流量的报文长度在报文头末尾填充相应数据 0，然后将报文输出给接口输出处理模块。在流量调度发送过程中，测试仪控制器可对 8 条流量的报文头进行更新，流量生成模块检测到报文头更新完成信号后，将调度发送更新后的报文头。

接口输出处理模块将测试报文的位宽由 134bit 转换为 8bit，以及将报文传输跨时钟域由测试仪内部逻辑工作时钟域切换到 GMII 发送时钟域，然后数据从 GMII1 接口输出。

2.2.7 测试报文接收处理流程

从网络中返回测试仪的测试报文由 GMII2 接口输入，在 GMII2 接口接收处理模块先对报文进行跨时钟域（从 GMII 接收时钟域到测试仪内部逻辑工作时钟域）转换，接下来将帧每拍数据的位宽由 8bit 转换为 134bit 后，增加两拍 metadata，在第一拍 metadata 中携带帧长度和接口接收帧的时间戳，然后将帧输出给流量统计与采样模块。

流量统计模块接收到测试报文，提取出报文的五元组信息，并与测试仪控制器配置的五元组进行匹配（带掩码），统计每条流量接收的报文个数；同时统计从 GMII2 接口接收的报文总数，并对接收到的报文进行封装，在报文封装时，会检测封装后的报文是否超过 1514B（不包括 4B 的 CRC），并将超出 1514B 的数据丢弃，以保证封装后报文长度不超过 1514B；然后按照配置的采样频率对接收的报文进行采样，将采样后的报文输出给数据选择器。

数据选择器先将采样报文写入 fifo 中进行缓存，在配置与状态管理模块无上报请求且无报文传输时，将 fifo 中的报文读出并输出给接口输出处理模块。

接口输出处理模块将测试报文的位宽由 134bit 转换为 8bit，以及将报文传输跨时钟域由测试仪内部逻辑工作时钟域切换到 GMII 发送时钟域，然后数据从 GMII3 接口输出。

3、详细设计

3.1 输入接口（IIP）模块设计

3.1.1 功能分析

IIP 模块主要功能：

1）去掉以太网帧首部的前导码、帧开始定界符；

2）跨时钟域处理，使用异步 FIFO(位宽 9bit、深度 16)来实现时钟域的处理。

3）记录时间同步报文的接收时间戳。

4）转换报文的位宽。

3.1.2 内部功能划分

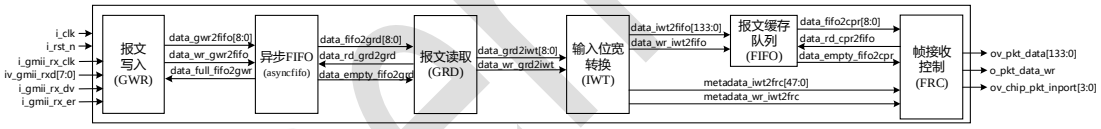

图 3-1 IIP 模块组成框图

GWR（Gmii write，GMII 接口写数据）模块：主要功能是将接收的 8bit Gmii 数据转换成 9bit 带首尾标识的 pkt_data；并写入异步 FIFO 中以便进行跨时钟处理。

GRD（Gmii read，GMII 接口读数据）模块：主要功能是在系统时钟下从异步 FIFO 中读取和输出 9bit 带首尾标识的 pkt_data，并在记录和输出第 1 拍时的接收时间戳。

IWT(input_width_transform，分组数据位宽转换)模块:主要功能将位宽为 8bit 的分组数据转换成位宽为 134bit 的分组数据进行传输；

PRD(pkt_read，报文读取)模块：将转换好的报文分组从 fifo 中读出。

3.1.3 信号定义

IIP 模块的接口信号的具体含义如表 3-1 所示。

表 3-1 IIP 模块接口信号含义

接口信号	位宽	方向	含义
i_clk	1	input	125MHz 的时钟
i_rst_n	1	input	复位，低有效
i_gmii_rst_n	1	input	GMII 接口复位，低有效
i_timer_rst	1	input	时钟复位信号
iv_syned_global_time	48	input	全局时钟信号
clk_gmii_rx	1	input	GMII 接收时钟，125MHz
i_gmii_dv	1	input	GMII 接口接收数据的写信号
iv_gmii_rxd	8	input	GMII 接口接收数据
i_gmii_er	1	input	GMII 接口接收错误信号
ov_data	134	output	输出分组数据
ov_relative_time	19	output	硬件获取的时间戳信号
o_data_wr	1	output	输出分组数据有效信号
o_fifo_overflow_pulse	1	output	fifo 溢出信号，当 fifo 写满时，该信号置位 1
o_fifo_underflow_pulse	1	output	当 fifo 被读空时，该脉冲置 1
内部信号			
data_gwr2fifo	9	GWR2FIFO	写到异步 FIFO 的数据
data_wr_gwr2fifo	1	GWR2FIFO	写到异步 FIFO 的数据写信号
data_full_fifo2gwr	1	FIFO2GWR	异步 FIFO 满信号
data_fifo2grd	9	FIFO2PRP	异步 FIFO 输出的数据
data_rd_grd2fifo	1	GRD2FIFO	异步 FIFO 的数据读信号
data_empty_fifo2grd	1	FIFO2GRD	异步 FIFO 空信号
wv_relative_time_timer2grd	19	TIMER2GRD	时钟信息
wv_relative_time_grd2iwt	19	GRD2IWT	时钟信息

接口信号	位宽	方向	含义
wv_global_time_grd2iwt	48	GRD2IWT	全局时钟信息
data_grd2iwt	134	GRD2IWT	GRD 模块输出的数据
data_wr_grd2iwt	1	GRD2IWT	GRD 模块输出的数据写信号
IWT-FIFO 信号定义			
data_iwt2fifo	134	IWT2FIFO	写到 FIFO 中的数据
data_wr_grd2iwt	1	IWT2FIFO	写到 FIFO 中的数据写信号
wv_time_length_iwt2fifo	31	IWT2FIFO	写到 FIFO 中的数据
w_time_length_wr_iwt2fifo	1	IWT2FIFO	写到 FIFO 中的数据写信号
wv_time_length_fifo2frc	31	FIFO2PRD	FIFO 输出的数据
w_time_length_fifo_rd_frc2fifo	1	FIFO2PRD	FIFO 的数据读信号
w_time_length_fifo_empty_fifo2frc	1	FIFO2PRD	FIFO 空信号
FIFO-PRD 信号定义			
data_fifo2frc	134	FIFO2FRC	FIFO 输出的数据
data_rd_frc2fifo	1	FRC2FIFO	FIFO 的数据读信号
data_empty_fifo2frc	1	FIFO2FRC	FIFO 空信号

3.1.4. 处理流程

■GWR 模块的处理流程

在 GMII 接口接收时钟 clk_gmii_rx 下，需要将输入信号通过寄存器缓存 1 拍，以便判断报文尾位置。

状态机包含 3 个状态。

初始状态 start_s：根据 i_gmii_dv 使能信号判断 GMII 接口接收数据信号是否有效，当 i_gmii_dv 高有效时，需要延时 1 拍，在下一拍进入开始写 FIFO 的 trans_s 状态，当 i_gmii_dv 为低时，状态机保持在 start_s 状态；

正常接收状态 trans_s：在写数据到 FIFO 时，首先判断 fifo 是否

已满，如果 fifo 已满，并且当前又是最后一拍，则跳转到 start_s 状态，如果不是最后一拍，则跳转到 full_error_s 状态；如果 fifo 未满，当前拍是最后一拍传输的数据，则跳转到 start_s 状态，否则状态机一致处于 trans_s 状态；

溢出错误状态 full_error_s：当 i_gmii_dv 高有效时，状态机一直处于 full_error_s 状态，否则状态机一跳回 start_s 状态。

■GRD 模块的处理流程

报文处理流程采用状态机实现，状态机包含 5 个状态。

初始状态 idle_s：在系统时钟 clk_sys 下，判断 fifo 是否为空，如果非空，则延迟 3 拍开始进入 head_s 状态读取数据，如果为空，则一直在 idle_s 状态。

识别报文头状态 head_s：在读取数据时，首先传输第一拍，根据输入每拍的最高位判断是否为头拍（最高位置 1 表示头拍和尾拍）且 fifo 是否不空。若是头拍且不空，根据端口类型、配置完成寄存器以及报文类型的状态，进入 tran_s 状态和 discard_s 状态可传输不同类型的报文；如果不是头拍或者 fifo 为空，状态机跳回初始状态。

正常传输状态 tran_s：根据输入每拍的最高位判断，若不是尾拍且 fifo 不为空，停留在 tran_s 状态进行报文正常传输；若是尾拍数据，报文传输结束，则给出报文有效脉冲信号，并且状态跳回 idle_s；如果在传输过程中出现 fifo 为空但又没有读出最后一拍时，表明报文已经断掉，则发出 fifo 为空的脉冲，状态跳转至 rdempty_error_s。

丢弃状态 discard_s：根据输入每拍的最高位判断，如果是尾拍数

据，则跳回初始状态，否则一直停留在 discard_s 状态将 fifo 中的数据读出丢弃；

读空错误状态 rdempty_error_s：根据输入每拍的最高位判断，如果是尾拍数据，则跳回初始状态，否则一直停留在 rdempty_error_s 状态将 fifo 中的数据读出丢弃；

■TIMER 模块的处理流程

在系统时钟 clk_sys 下，对时钟进行计数，得到绝对时间；当 timer_rst 为高有效时，将时钟计数清零；当计数到 4ms 时（计数器的数值为拍数，当计数到 0x7A120 拍时，代表 4ms），该计数器清零。

■IWT 模块的处理流程

将位宽为 8bit 的分组数据转换为位宽为 134bit 的报文分组数据，并将转换好的报文传输至下一级模块。

报文处理流程采用状态机实现，状态机包括三个状态。

初始状态 IDLE_S：等待报文分组的首字节数据，否则一直在初始状态。

发送 metadata 状态 TRANS_MD1_S：当跳入这个状态，直接发送一拍 metadata 数据，然后无条件进入下一个状态。

等待数据分组转换完成状态 WRITE_REG_S：进行位宽 8-128 的转换，分组尾部不足 16 拍的补齐 16 拍进行转换。

■PRD 模块的处理流程

将报文分组数据从 fifo 中读出。

报文处理流程采用状态机实现，状态机包括三个状态。

初始状态 IDLE_S：根据缓存时间信息的 fifo 的 empty 信号，如果 empty 不空，则给出报文从 fifo 缓存区读出的读信号，并且跳转到下一个状态。

发送一拍 metadata 状态 TRANS_MD0_S：进入 TRANS_MD0_S 状态后，直接传输一拍 metadata 给输出，然后无条件跳转到下一状态。

数据分组传输状态 TRANS_DATA_S:识别报文尾拍数据，如果不是尾拍数据，一致在此状态传输，传输至尾拍，则跳转至初始状态。

3.2 数据分配器（DMUX）模块设计

3.2.1 功能分析

负责根据以太网类型对报文进行分派，将时间同步报文、时间同步封装报文分派给时间同步模块 CSC，配置报文（包括配置时间同步相关参数、测试报文特征相关参数）分派给配置与状态管理模块 CSM。

3.2.2 内部功能划分

根据 DMUX 模块的功能分析，在 DMUX 模块的内部无需进一步划分。DMUX 模块的接口信号如图 3-2 所示。

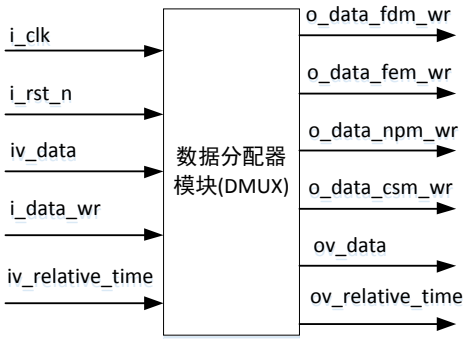

图 3-2 DMUX 模块组成框图

3.2.3 信号定义

数据分配器(DMUX)模块的信号含义如表 3-2 所示。

表 3-2. DMUX 模块接口信号含义

接口信号	位宽	方向	含义
i_clk	1	input	输入时钟信号
i_rst_n	1	input	复位信号
iv_data	134	input	输入分组数据
iv_relative_time	19	input	记录的时间戳信号输入
i_data_wr	1	input	输入分组数据有效信号
ov_relative_time	19	output	记录的时间戳信号输出
ov_data	134	output	输出分组数据
o_data_csm_wr	1	output	输出至 csm 模块分组数据有效信号
o_data_npm_wr	1	output	输出至 npm 模块分组数据有效信号
o_data_fem_wr	1	output	输出至 fem 模块分组数据有效信号
o_data_fdm_wr	1	output	输出至 fdm 模块分组数据有效信号

3.2.4 处理流程

■DMUX 模块的处理流程

将不同类型的报文数据进行分派。将 PTP 帧、NMAC 状态上报帧，分派给帧封装模块进行封装；将芯片配置帧、ptp 封装帧分派给帧解封装模块进行解封装处理。

根据报文类型对报文进行分配，通过 1 个状态机实现，包括以下 6 个状态。具体状态机设计如下。

初始状态 IDLE_S：判断分组数据是否为高，并且分组是不是第一拍，若是,再判断报文是不是 TSMP 帧并且报文类型是不是配置报文，如果是，将报文转发 csm 模块；若是状态上报报文，则报文转发至 npm 模块，若是 ptp 报文，则将报文转发至 FEM 模块，若是 ptp

封装报文，则将报文转发至 fdm 模块，否则进入丢弃状态；

报文传输至 csm 模块状态 TRANS_TO_CSM_S：根据尾拍标识位判断，直至报文传输完成才跳转回初始状态；

报文传输至 npm 模块状态 TRANS_TO_NPM_S：根据尾拍标识位判断，直至报文传输完成才跳转回初始状态；

报文传输至 fem 模块状态 TRANS_TO_FEM_S：根据尾拍标识位判断，直至报文传输完成才跳转回初始状态；

报文传输至 fdm 模块状态 TRANS_TO_FDM_S：根据尾拍标识位判断，直至报文传输完成才跳转回初始状态；

报文丢弃状态 DISC_DATA_S：根据尾拍标识位判断，直至报文丢弃完成才跳转回初始状态。

3.3 配置与状态管理(CSM)模块设计

3.3.1 功能分析

负责解析配置报文（包括配置时间同步相关参数、测试报文的特征参数），完成对本地寄存器和表的配置；以及周期性上报本地的状态信息。

3.3.2 内部功能划分

根据 CSM 模块的功能分析，CSM 模块划分的功能组成框图如图 3-3 所示

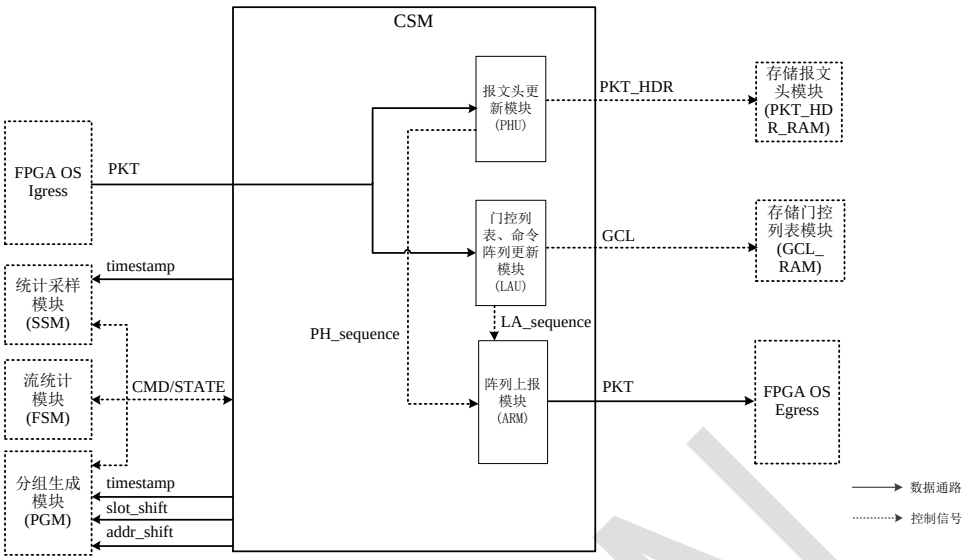

图 3-3 CSM 模块组成框图

PHU(PKT Header Update，报文头更新模块)模块：在收到来自 FPGA OS Igress 的分组后，判断其是否为用来更新报文头的 Beacon 分组：若是，则将报文头读出，写到 PKT_HDR_RAM 中，并且将该分组的序列号传给 ARM 模块；若不是，则将该分组丢弃。

LAU(List and Array Update，门控列表和命令阵列更新模块)模块：在收到来自 FPGA OS Igress 的分组后，判断其是否为用来更新门控列表和命令阵列的 Beacon 分组：若是，则将门控列表读出，写到 GCL_RAM 中，并且将命令阵列读出，更新模块内对应寄存器的值，同时将该分组的序列号传给 ARM 模块；若不是，则将该分组丢弃。

ARM(Array Report Module，阵列上报模块)模块：本地时钟计数器值每经过 1s，按照附录二的 Beacon Report 报文格式定义构造一个 Beacon Report 分组，发送给软件，用来上报当前 FAST UM 的全部状态信息。

3.3.3 信号定义

配置与状态管理(CSM)模块的信号含义如表 3-3 所示。

表 3-3. CSM 模块接口信号含义

接口信号	位宽	方向	含义
clk	1	input	输入时钟信号
rst_n	1	input	复位信号
o_reg_rst	1	output	寄存器复位信号
iv_dmac	48	input	输入目的 MAC
iv_smac	48	input	输入源 MAC
i_time_slot_switch	1	input	时间槽开关信号
iv_time_slot	10	input	时间槽
iv_timestamp	48	input	时钟
i_report_pulse	1	input	上报脉冲信号
in_lcm_data	134	input	输入分组数据
in_lcm_data_wr	1	input	输入分组数据有效信号
o_data_lcm2mux_req	1	output	状态报文数据传输请求信号
i_data_mux2lcm_ack	1	input	状态报文数据传输响应信号
ov_data_lcm2mux	134	output	状态报文数据传输分组
out_lcm_pkt_1_len	12	output	类型 1 报文长度
out_lcm_pkt_2_len	12	output	类型 2 报文长度
out_lcm_pkt_3_len	12	output	类型 3 报文长度
out_lcm_pkt_4_len	12	output	类型 4 报文长度
out_lcm_pkt_5_len	12	output	类型 5 报文长度
out_lcm_pkt_6_len	12	output	类型 6 报文长度
out_lcm_pkt_7_len	12	output	类型 7 报文长度
out_lcm_pkt_8_len	12	output	类型 8 报文长度
out_lcm_tb_1_size	16	output	类型 1 报文令牌桶大小
out_lcm_tb_1_rate	16	output	类型 1 报文令牌桶速率
out_lcm_tb_2_size	16	output	类型 2 报文令牌桶大小
out_lcm_tb_2_rate	16	output	类型 2 报文令牌桶速率
out_lcm_tb_3_size	16	output	类型 3 报文令牌桶大小
out_lcm_tb_3_rate	16	output	类型 3 报文令牌桶速率

接口信号	位宽	方向	含义
out_lcm_tb_4_size	16	output	类型 4 报文令牌桶大小
out_lcm_tb_4_rate	16	output	类型 4 报文令牌桶速率
out_lcm_tb_5_size	16	output	类型 5 报文令牌桶大小
out_lcm_tb_5_rate	16	output	类型 5 报文令牌桶速率
out_lcm_tb_6_size	16	output	类型 6 报文令牌桶大小
out_lcm_tb_6_rate	16	output	类型 6 报文令牌桶速率
out_lcm_tb_7_size	16	output	类型 7 报文令牌桶大小
out_lcm_tb_7_rate	16	output	类型 7 报文令牌桶速率
out_lcm_tb_8_size	16	output	类型 8 报文令牌桶大小
out_lcm_tb_8_rate	16	output	类型 8 报文令牌桶速率
out_lcm_rule_5tuple_1	104	output	类型 1 报文五元组规则
out_lcm_mask_1	104	output	类型 1 报文掩码
out_lcm_rule_5tuple_2	104	output	类型 2 报文五元组规则
out_lcm_mask_2	104	output	类型 2 报文掩码
out_lcm_rule_5tuple_3	104	output	类型 3 报文五元组规则
out_lcm_mask_3	104	output	类型 3 报文掩码
out_lcm_rule_5tuple_4	104	output	类型 4 报文五元组规则
out_lcm_mask_4	104	output	类型 4 报文掩码
out_lcm_rule_5tuple_5	104	output	类型 5 报文五元组规则
out_lcm_mask_5	104	output	类型 5 报文掩码
out_lcm_rule_5tuple_6	104	output	类型 6 报文五元组规则
out_lcm_mask_6	104	output	类型 6 报文掩码
out_lcm_rule_5tuple_7	104	output	类型 7 报文五元组规则
out_lcm_mask_7	104	output	类型 7 报文掩码
out_lcm_rule_5tuple_8	104	output	类型 8 报文五元组规则
out_lcm_mask_8	104	output	类型 8 报文掩码
out_lcm_samp_freq	16	output	采样频率
in_pgm2lcm_pkt_1_cnt	32	input	类型 1 报文发送报文计数器
in_pgm2lcm_pkt_2_cnt	32	input	类型 2 报文发送报文计数器
in_pgm2lcm_pkt_3_cnt	32	input	类型 3 报文发送报文计数器
in_pgm2lcm_pkt_4_cnt	32	input	类型 4 报文发送报文计数器
in_pgm2lcm_pkt_5_cnt	32	input	类型 5 报文发送报文计数器

接口信号	位宽	方向	含义
in_pgm2lcm_pkt_6_cnt	32	input	类型 6 报文发送报文计数器
in_pgm2lcm_pkt_7_cnt	32	input	类型 7 报文发送报文计数器
in_pgm2lcm_pkt_8_cnt	32	input	类型 8 报文发送报文计数器
in_fsm2lcm_pkt_1_cnt	32	input	类型 1 报文接收报文计数器
in_fsm2lcm_pkt_2_cnt	32	input	类型 2 报文接收报文计数器
in_fsm2lcm_pkt_3_cnt	32	input	类型 3 报文接收报文计数器
in_fsm2lcm_pkt_4_cnt	32	input	类型 4 报文接收报文计数器
in_fsm2lcm_pkt_5_cnt	32	input	类型 5 报文接收报文计数器
in_fsm2lcm_pkt_6_cnt	32	input	类型 6 报文接收报文计数器
in_fsm2lcm_pkt_7_cnt	32	input	类型 7 报文接收报文计数器
in_fsm2lcm_pkt_8_cnt	32	input	类型 8 报文接收报文计数器
in_ssm2lcm_pkt_cnt	32	input	采样报文计数器
out_lcm_addr_shift	1	output	地址输出信号
out_lcm_test_start	1	output	测试开始信号
o_lau_update_finish	1	output	表项更新完成信号
out_lcm_pkt_hdr_wr	1	output	报文头输出使能信号
out_lcm_pkt_hdr_addr	6	output	报文头输出地址信号
out_lcm_pkt_hdr	128	output	输出报文头数据
out_lcm_gc_wr	1	output	输出门控使能信号
out_lcm_gc_addr	5	output	输出门控地址信号
out_lcm_gc	128	output	输出门控数据
内部信号			
phu_update_finish	1	lcm2phu	更新报文头完成信号
pkt_hdr_seq	8	arm2phu	8 种报文头配置成功信号
gcl_array_seq	4	lau2phu	门控阵列配置成功信号

3.3.4 处理流程

■PHU 模块的处理流程

PHU 模块功能是在收到来自 FPGA OS Igress 的分组后，判断其是否为用来更新报文头的 Beacon 分组：若是，则将报文头读出，写

到 PKT_HDR_RAM 中，并且将该分组的序列号传给 ARM 模块；若不是，则将该分组丢弃。具体状态机实现如下：

初始状态 IDLE_S：等待分组传来，若分组第一拍传来（数据使能有效信号为 1 并且数据位[133:132]==2'b01），则跳转到 WAIT1_S。

等待第一拍状态 WAIT1_S：这一拍为 metadata1，无条件地跳转到 JUDGE_TYPE_S。

判断分组类型状态 JUDGE_TYPE_S:若该分组的以太网类型为 0xff01 且消息类型为 0x1(in_phu_data[15:12]==4'h1)，则说明该分组为用来更新报文头的 Beacon 分组，跳转到 WAIT2_S；否则跳转到 IDLE_S。

等待第二拍状态 WAIT2_S：这一拍为被封装的 FAST 头的 metadata0，无条件地跳转到 WAIT3_S。

等待第三拍状态 WAIT3_S：这一拍为被封装的 FAST 头的 metadata1，无条件地跳转到 READ_PKT_HDR_S。

读出报文头第一拍状态 READ_PKT_HDR1_S：在此状态读出报文头第一拍，并写到 PKT_HDR_RAM：将报文头数据写信号置 1（out_phu_pkt_hdr_wr <= 1'b1），报文头数据第一拍输出（out_phu_pkt_hdr <= in_phu_data[127:0]）；若往 PKT_HDR_RAM 高 32 位地址写报文头（out_phu_addr_shift==1'b1），则将写地址置为 6'd0（out_phu_pkt_hdr_addr<=6'd0），若往 PKT_HDR_RAM 低 32 位地址写报文头（out_phu_addr_shift==1'b0），则将写地址置为 6'd32（out_phu_pkt_hdr_addr<=6'd32）；无条件地跳转到

READ_PKT_HDR_S。

读出报文头状态 READ_PKT_HDR_S：在此状态读出报文头的其余 31 拍数据，并写到 PKT_HDR_RAM，若该拍为报文中间数据（ in_phu_data[133:132]==2’b11 ），则将数据写地址加 1（out_phu_pkt_hdr_addr<= out_phu_pkt_hdr_addr+6’d1），报文头数据输出（ out_phu_pkt_hdr <= in_phu_data[127:0] ），留在 READ_PKT_HDR_S ；若该拍为报文最后一拍（ in_phu_data[133:132]==2’b10 ），则将数据写信号置 0（ out_phu_pkt_hdr_wr <= 1’b0 ），写地址置 0（ out_phu_pkt_hdr_addr<=6’d0 ），报文头数据输出置 0（out_phu_pkt_hdr <= 128’b0），往 PKT_HDR_RAM 高/低 32 位地址写报文头信号取非（out_phu_addr_shift<=~out_phu_addr_shift），报文头更新完成信号置 1（out_phu_update_finish==1’b1），跳转到 IDLE_S。

■LAU 模块的处理流程

LAU 模块功能是在收到来自 FPGA OS Igress 的分组后，判断其是否为用来更新门控列表和命令阵列的 Beacon 分组：若是，则将门控列表读出，写到 GCL_RAM 中，并且将命令阵列读出，更新模块内对应寄存器的值，同时将该分组的序列号传给 ARM 模块；若不是，则将该分组丢弃。状态机实现如下。

空闲状态 IDLE_S：等待分组传来，若分组第一拍传来（in_lau_data_wr == 1'b1 && in_phu_data[133:132]==2'b01），则跳转

到 WAIT1_S。

等待第一拍状态 WAIT1_S：这一拍为 metadata1，无条件地跳转到 JUDGE_TYPE_S。

判断分组类型状态 JUDGE_TYPE_S：若该分组的以太网类型为 0xff01(in_phu_data[31:16]==16'hff01) 且消息类型为 0x2(in_phu_data[15:0]==16'h2)，则说明该分组为用来更新门控列表、命令阵列的 Beacon 分组，跳转到 WAIT2_S；否则跳转到 IDLE_S。

等待第二拍状态 WAIT2_S：这一拍为被封装的 FAST 头的 metadata0，无条件地跳转到 WAIT3_S。

等待第三拍状态 WAIT3_S：这一拍为被封装的 FAST 头的 metadata1，无条件地跳转到 READ_GCL_S。

读出门控列表状态 READ_GCL_S：在此状态读出 32 拍门控列表数据(详见附录二 Beacon Update GCL_CA 报文消息域划分表)，并写到 GCL_RAM；在模块内维护一个用来统计 32 拍门控列表已读出拍数的计数器 reg_gcl_cnt，若门控列表未全部读出（reg_gcl_cnt<5'd31），则留在 READ_GCL_S；若门控列表全部读出（reg_gcl_cnt==5'd31），则跳转到 READ_PGM_S。

读出 PGM 模块的命令阵列状态 READ_PGM_S：在此状态读出 6 拍 PGM 模块的命令阵列数据(详见附录二 Beacon Update GCL_CA 报文消息域划分表)，并更新对应寄存器的值；其中对软件配置的 8 种报文长度会进行判断：若配置的报文长度≤12'd60，则将 12'd60 赋给相应的报文长度寄存器；若配置的报文长度≥12'd1514，则将 12'd1514

赋给相应的报文长度寄存器；否则直接将软件配置的值赋给相应的报文长度寄存器。在模块内维护一个用来统计 6 拍 PGM 模块的命令阵列已读出拍数的计数器 reg_pgm_cnt，若 PGM 模块的命令阵列未全部读出（reg_pgm_cnt<4’d5），则留在 READ_PGM_S；若 PGM 模块的命令阵列全部读出（reg_gcl_cnt==4’d5），则跳转到 READ_FSM_S。

读出 FSM 模块的命令阵列状态 READ_FSM_S：在此状态读出 17 拍 FSM 模块的命令阵列数据(详见附录二 Beacon Update GCL_CA 报文消息域划分表)，并更新对应寄存器的值；在模块内维护一个用来统计 17 拍 FSM 模块的命令阵列已读出拍数的计数器 reg_fsm_cnt，若 FSM 模块的命令阵列未全部读出（reg_fsm_cnt<5’d16），则留在 READ_FSM_S；若 FSM 模块的命令阵列全部读出（reg_gcl_cnt==5’d16），则跳转到 READ_SSM_S。

读出 SSM 模块的命令阵列状态 READ_SSM_S：在此状态读出 1 拍 SSM 模块的命令阵列数据(详见附录二 Beacon Update GCL_CA 报文消息域划分表)，并更新对应寄存器的值，将测试开始/停止信号取非并赋给门控列表、命令阵列更新完成信号，无条件跳转到 IDLE_S。

■ARM 模块的处理流程

ARM 模块功能是本地时钟计数器值每经过 1s，按照附录二的 Beacon Report 报文格式定义构造一个 Beacon Report 分组，发送给软件，用来上报当前 UM 的全部状态信息；当收到测试停止信号后，经过 1s 再上报一次，然后停止上报。状态机实现如下。

初始状态 IDLE_S：若本地时钟计数器值经过周期 1s，生成

metadata0，跳转到 GENERATE_MD1_S；若测试停止信号传来，且本地时钟计数器值经过 1s800ns 后，停留在 IDLE_S。

生成 metadata1 状态 GENERATE_MD1_S：生成 metadata1，无条件地跳转到 GENERATE_ETH_HDR_S。

生成以太网帧头状态 GENERATE_ETH_HDR_S：生成以太网帧头，其中目的 MAC 为 0xffff_ffff_ffff，源 MAC 为 0x0000_0000_0000，长度类型域为 0xff01，消息类型域为 0x3，并填上配置门控列表、命令阵列的报文序列号和配置报文头的报文序列号，无条件地跳转到 GENERATE_ENCAP_MD0_S。

生成被封装的 metadata0 状态 GENERATE_ENCAP_MD0_S：生成被封装的 metadata0，无条件地跳转到 GENERATE_ENCAP_MD1_S。

GENERATE_ENCAP_MD1_S：生成被封装的 metadata1 状态。生成被封装的 metadata1，无条件地跳转到 GENERATE_PGM_CA_S。

生成 PGM 模块的命令阵列状态 GENERATE_PGM_CA_S：PGM 模块的命令阵列总共有 6 拍；在模块内部维护一个计数器 reg_pgm_ca_cnt，若 reg_pgm_ca_cnt<4’d5，则留在 GENERATE_PGM_CA_S；若 reg_pgm_ca_cnt==4’d5，则跳转到 GENERATE_FSM_CA_S。

生成 FSM 模块的命令阵列状态 GENERATE_FSM_CA_S：FSM 模块的命令阵列总共有 17 拍；在模块内部维护一个计数器 reg_fsm_ca_cnt，若 reg_fsm_ca_cnt<5’d16，则留在 GENERATE_FSM_CA_S；若 reg_fsm_ca_cnt==5’d16，则跳转到

GENERATE_SSM_CA_S。

生成 SSM 模块的命令阵列状态 GENERATE_SSM_CA_S：SSM 模块的命令阵列总共有 2 拍；在模块内部维护一个计数器 reg_ssm_ca_cnt，若 reg_ssm_ca_cnt<4'd1，则留在 GENERATE_SSM_CA_S；若 reg_ssm_ca_cnt==4'd1，则跳转到 GENERATE_PGM_SA_S。

生成 PGM 模块的状态阵列状态 GENERATE_PGM_SA_S：PGM 模块的状态阵列总共有 3 拍；在模块内部维护一个计数器 reg_pgm_sa_cnt，若 reg_pgm_sa_cnt<4'd2，则留在 GENERATE_PGM_SA_S；若 reg_pgm_sa_cnt==4'd2，则跳转到 GENERATE_FSM_SA_S。

生成 FSM 模块的状态阵列状态 GENERATE_FSM_SA_S：FSM 模块的状态阵列总共有 3 拍；在模块内部维护一个计数器 reg_fsm_sa_cnt，若 reg_fsm_sa_cnt<4'd2，则留在 GENERATE_FSM_SA_S；若 reg_fsm_sa_cnt==4'd2，则跳转到 GENERATE_SSM_SA_S。

生成 SSM 模块的状态阵列状态 GENERATE_SSM_SA_S：SSM 模块的状态阵列总共有 1 拍，无条件跳转到 IDLE_S

3.4 流量生成（TGE）模块设计

3.4.1 功能分析

TGE 模块负责生成多条指定速率的不同类型的并发流量，需要

支持以下功能：

1) 支持生成发送 8 种类型的报文；
2) 支持基于令牌桶机制进行限速；
3) 支持基于时间感知调度器（TAS）来调度数据；
4) 支持在 8 个数据队列的门为开的队列中按照绝对优先级进行调度；
5) 支持软件在测试过程中动态更新 8 种类型的报文头
6) 支持在用户配置的报文头的基础上进行报文扩展，增加报文发送时间戳、序列号等字段；
7）支持同 CSM 模块进行通信，从而实现对不同参数寄存器的更新与上报、复位。

3.4.2 内部功能划分

根据 TGE 模块的功能分析，对 TGE 模块的内部功能划分，组成框图如图 3-4 所示。

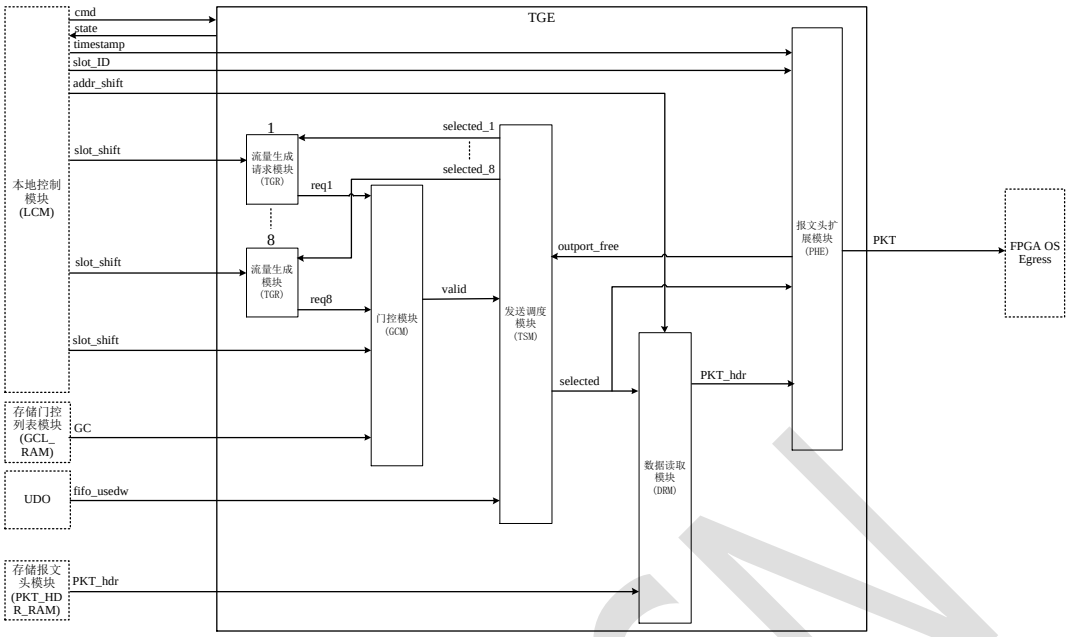

图 3-4 TGE 模块组成框图

TGR(Traffic Generation Request，流量生成请求模块)模块：基于令牌桶机制进行流量限速。每经过一个时间槽往令牌桶中添加一定数量的令牌，每个令牌代表 1B，当令牌桶中剩余令牌数≥报文字节数+4 字节（CRC）时，则该种类型报文有流量生成请求，当该类型报文被调度发送时，将令牌桶中剩余令牌数减去该报文字节数和 4 字节（CRC）。

GCM(Gate Control Module，门控模块)模块：在门控列表、命令阵列更新完成后（此时测试还未开始），GCM 模块从 GCL RAM 中读出一拍 128 位的门控状态；在门控列表、命令阵列和报文头都更新完成后每隔 16 个时间槽，GCM 模块从 GCL RAM 中读出 128 位的门控状态；若在当前时间槽某类型报文对应的门控为开，则该类型报文的请求有效；否则无效。在收到测试结束信号时，将传给发送调度模块的有效信号全部置 0。

TSM(Transmitting and Scheduling Module，发送调度模块)模块：在刚复位完成或端口空闲，根据接收到的 8 位流量生成请求有效信号，按照优先级的顺序进行调度，并把调度结果传给 TGR、DRM 和 PHE 模块。

DRM(Data Reading Module，数据读取模块)模块：根据从 TSM 传来的调度结果及 LCM 传来的两组报文头切换信号将相应的报文头的读信号及地址传给 PKT_HDR_RAM；当收到报文头后，转发给 PHE 模块。

PHE(PKT_header Extension，报文头扩展模块)模块:根据 DRM 传来的报文头、LCM 传来的分组长度、TSM 传来的调度结果生成分组：增加 2 拍的 metadata，在 128 位数据前面增加 6 位（2 位的头尾标识、4 位无效字节数）；在报文头后面增加发送时间戳、报文序列号等字段，并在剩余的空闲字段填充 0。统计 8 种类型报文的发送个数。

3.4.3 信号定义

TGE 模块的信号含义如表 3-4 所示。

表 3-4 TGE 模块接口信号含义

信号名	位宽	方向	备注
clk	1	input	125Mhz 的时钟信号
rst_n	1	input	复位信号，低有效
PGM-LCM 信号			
timestamp	48	input	来自 LCM 模块的时间戳
pkt_N_len	12	input	第 N 种类型报文的字节数，N=1、2、…、8

信号名	位宽	方向	备注
out_pgm_pkt_N_cnt	64	output	第 N 种类型报文的发送个数，N=1、2、…、8
in_pgm_addr_shift	1	input	PKT_HDR_RAM 内两组 8 种报文头存储地址块选择信号；0 表示读取 RAM 低 32 位地址，1 表示读取 RAM 高 32 位地址
in_pgm_slot_shift	1	input	时间槽切换信号；一个时间槽结束时置为 1
slot_ID	9	input	当前时间槽 ID（取值 0、1、…、511）
in_pgm_tb_N_size	16	input	第 N 种类型报文对应的令牌桶容量，N=1、2、…、8
in_pgm_tb_N_rate	16	input	第 N 种类型报文对应的单位时间槽往令牌桶中增加令牌的数量，N=1、2、…、8
in_pgm_test_start	1	input	测试开始信号
in_pgm_test_stop	1	input	测试结束信号
cnt_rst	1	input	寄存器/计数器复位信号，高有效
PGM-PKT_HDR_RAM 信号			
out_pgm_pkt_hdr_rd	1	output	从 RAM 中读取报文头的读信号
out_pgm_pkt_hdr_addr	6	output	从 RAM 中读取报文头的地址
in_pgm_pkt_hdr	128	input	从 RAM 中读取的报文头数据
PGM-GCL_RAM 信号			
out_pgm_gcl_rd	1	output	从 RAM 中读取门控状态的读信号
out_pgm_gcl_addr	5	output	从 RAM 中读取门控状态的地址
in_pgm_gc	128	input	16 个时间槽的 8 种类型报文的门控状态
PGM-UDO 信号			
fifo_usedw	8	input	UDO 中 fifo 计数值
PGM-FPGA OS Egress 信号			
out_pgm_data	134	output	PGM 模块输出的数据
out_pgm_data_wr	1	output	PGM 模块输出的数据的写信号
out_pgm_data_valid	1	output	PGM 模块输出的报文的有效信号
out_pgm_data_valid_wr	1	output	PGM 模块输出的报文的有效信号的写信号
内部信号			

信号名	位宽	方向	备注
tgr2gcm_req_1	1	tgr2gcm	类型 1 报文生成请求是否有效信号
tgr2gcm_req_2	1	tgr2gcm	类型 1 报文生成请求是否有效信号
tgr2gcm_req_3	1	tgr2gcm	类型 1 报文生成请求是否有效信号
tgr2gcm_req_4	1	tgr2gcm	类型 1 报文生成请求是否有效信号
tgr2gcm_req_5	1	tgr2gcm	类型 1 报文生成请求是否有效信号
tgr2gcm_req_6	1	tgr2gcm	类型 1 报文生成请求是否有效信号
tgr2gcm_req_7	1	tgr2gcm	类型 1 报文生成请求是否有效信号
tgr2gcm_req_8	1	tgr2gcm	类型 1 报文生成请求是否有效信号
gcm2tsm_valid	8	gcm2tsm	门控有效信号
drm2phe_pkt_hdr	128	drm2phe	报文头分组数据
drm2phe_pkt_hdr_wr	1	drm2phe	报文头分组数据有效信号

3.4.4 处理流程

■TGR 模块模块的处理流程

TGR 模块功能是基于令牌桶机制进行流量限速。每经过一个时间槽往令牌桶中添加一定数量的令牌，每个令牌代表 1B，当令牌桶中剩余令牌数≥报文字节数+4 字节（CRC）时，则该种类型报文有流量生成请求，当该类型报文被调度发送时，将令牌桶中剩余令牌数减去该报文字节数和 4 字节（CRC）。

由于每种类型的流量都需要使用令牌桶机制进行流量限速，所以 TGR 模块需要实例化 8 个。

状态机实现包括两个状态。

初始赋值状态 INIT_S：等待门控列表、命令阵列更新完成；若更新完成，则对令牌桶中剩余令牌 remain_tokens 进行赋值（remain_tokens<= in_tgr_tb_rate），跳转到 TB_UPDATA_S；否则停留在 INIT_S。

令牌桶中剩余令牌更新状态 TB_UPDATA_S：若测试未停止，则停留在 TB_UPDATA_S 状态；若该类型报文头被调度发送（in_tgr_select==1'b1），将该报文长度（字节）+4 字节（CRC）赋给需消耗的令牌数 consume_tokens（consume_tokens<= in_tgr_pkt_len+12'd4），否则将 consume_tokens 置 0（consume_tokens<=12'd0）；对令牌桶中剩余令牌进行更新：若令牌数量未超过桶的容量（remain_tokens + in_tgr_tb_rate –consume_tokens < in_tgr_tb_size），则添加令牌后令牌桶中的剩余令牌为 remain_tokens + in_tgr_tb_rate - consume_tokens；若令牌数量超过桶的容量（remain_tokens + in_tgr_tb_rate –consume_tokens ≥ in_tgr_tb_size），则添加令牌后令牌桶中的剩余令牌数为桶的容量 in_tgr_tb_size；若令牌桶中剩余令牌数≥报文长度（字节）+4 字节（CRC）（remain_tokens ≥in_tgr_pkt_len+12'd4），说明该类型的流量有生成请求，将流量生成请求信号置 1（out_tgr_req<=1'b1），否则将 out_tgr_req 置 0。若测试停止，则将令牌桶中剩余令牌进行复位清零，跳转到 INIT_S。

■GCM 模块模块的处理流程

GCM 模块功能是在门控列表、命令阵列更新完成后（此时测试还未开始），GCM 模块从 GCL RAM 中读出一拍 128 位的门控状态；

在门控列表、命令阵列和报文头都更新完成后每隔 16 个时间槽，GCL 模块从 GCL RAM 中读出 128 位的门控状态；若在当前时间槽某类型报文对应的门控为开，则该类型报文的请求有效；否则无效。在收到测试结束信号时，将传给发送调度模块的有效信号全部置 0。

状态机实现包括五个状态。

初始赋值状态 INIT_S：等待门控列表、命令阵列更新完成；若更新完成，则将从 RAM 中读取门控状态的读信号置 1（out_gcm_gcl_rd <= 1'b1），跳转到 WAIT1_S；否则 out_gcm_gcl_rd <= 1'b0，停留在 INIT_S。

空闲状态 IDLE_S：若测试停止信号未传来，将当前时间槽的 8 种类型流量的门控状态分别与对应的流量生成请求做与运算，得到 8 位的有效信号，并输出给 TSM 模块；当收到来自 LCM 模块的门控数据读取信号(gcl_ram_rd == 1'b1)，将从 RAM 中读取门控状态的读信号置 1（out_gcm_gcl_rd <= 1'b1），跳转到 WAIT1_S；若测试停止信号传来，将 8 位流量生成请求有效信号全部置 0（out_gcm_valid<=8'b0），输出给 TSM 模块，跳转到 INIT_S。

等待第一拍状态 WAIT1_S：读 GCL_RAM 数据生效需要在两拍后，因此，在 WAIT1_S 等待一拍，同时将当前时间槽的 8 种类型流量的门控状态分别与对应的流量生成请求做与运算，得到 8 位的有效信号，并输出给 TSM 模块；无条件地跳转到 WAIT2_S。

等待第两拍状态 WAIT2_S：将当前时间槽的 8 种类型流量的门控状态分别与对应的流量生成请求做与运算，得到 8 位的有效信号，

并输出给 TSM 模块；无条件地跳转到 READ_S。

读门控数据状态 READ_S：从 GCL_RAM 中读出一拍 128 位数据，将从 GCL_RAM 中读取门控状态的读信号置 0（out_gcm_gcl_rd<=0），从 GCL_RAM 中读取门控状态的地址加 1（out_gcm_gcl_addr <=out_gcm_gcl_addr + 5'd1）；同时将当前时间槽的 8 种类型流量的门控状态分别与对应的流量生成请求做与运算，得到 8 位的有效信号，并输出给 TSM 模块；无条件地跳转到 IDLE_S。。

■TSM 模块模块的处理流程

TSM 模块功能是在刚复位完成或端口空闲时，根据接收到的 8 位流量生成请求有效信号，按照优先级进行调度，并把调度结果传给 TGR、DRM 和 PHE 模块。

8 种类型的报文优先级：第 1 种类型报文>第 2 种类型报文>第 3 种类型报文>第 4 种类型报文>第 5 种类型报文>第 6 种类型报文>第 7 种类型报文>第 8 种类型报文。

为了使报文到达端口时，端口 fifo 有足够的空闲空间来存储报文且报文在 fifo 中等待发送的时间最短，需计算端口的 fifo 已用报文拍数 in_tsm_fifo_usedw[7:0]最小为多少时，TSM 才能调度报文。端口逻辑为当一个报文全部写入 fifo 后，才能开始读出该报文；每个时钟周期输出 8bit，传 128bit 需要 16 个时钟周期。从 in_tsm_fifo_usedw[7:0]达到阈值到被调度的报文（除去两拍 metadata 的）第一拍写入端口 fifo 需要 12 个时钟周期；当报文为 1518B（95 拍）时，从 in_tsm_fifo_usedw[7:0]达到阈值到该 1518B 报文全部写入端口 fifo 中

需要 95+12=107 个时钟周期，所以端口 fifo 已用报文拍数最小应该为 7 拍（107/16=6.7）；此时 fifo 剩余空闲空间为 128-7=121>95，足够存储一个最长的报文。根据 fifo 计数的特点，应将 in_tsm_fifo_usedw[7:0] 阈值设为 5。

状态机实现包括三个状态。

起始状态 IDLE_S：FPGA 上电或者上次调度的报文全部从 PGM 模块输出时，跳转到 UDO_FIFO_FREE_S。（初始标志信号 init_flag 作用：为了使 FPGA 上电时，队列能被调度，设置锁存器 init_flag，复位时置为 1。）

判断端口 fifo 是否有足够空闲空间状态 UDO_FIFO_FREE_S：端口 fifo 状态 fifo_usedw 阈值设置为 8'd5(详细分析见"3.2.3.4 c 端口 fifo_usedw 阈值的分析")，若 in_tsm_fifo_usedw<=8'd5，跳转到 PRIORITY_SCHEDULE_S；否则留在 UDO_FIFO_FREE_S。

按照优先级进行调度状态 PRIORITY_SCHEDULE_S：根据 GCM 模块传来的 8 种类型流量生成请求有效信号 in_tsm_valid，按照优先级进行调度，当 8 种类型流量有一种被选中（|out_tsm_selected==1'b1）时，跳转到 IDLE_S。

■DRM 模块的处理流程

DRM 模块功能是根据从 TSM 传来的调度结果及 LCM 传来的两组报文头切换信号将相应的报文头的读信号及地址传给 PKT_HDR_RAM；当收到报文头后，转发给 PHE 模块。

状态机实现包括六个状态。

初始状态 IDLE_S：若 8 种报文头至少有一个被选中（in_drm_selected[7:0]!=8'd0），则将从 PKT_HDR_RAM 中读取报文头的读信号置 1（out_drm_pkt_hdr_rd<=1'b1），并根据 PKT_HDR_RAM 内两组 8 种报文头存储地址块选择信号 in_drm_addr_shift 和被选中的报文头 in_drm_selected[7:0]给出数据读取地址：若从 PKT_HDR_RAM 的低 32 位地址读取数据（in_drm_addr_shift==1'b0），第 N 个报文头被选中（in_drm_selected[N]==1'b1），则将从 PKT_HDR_RAM 中读取报文头的地址置为 4*N（out_drm_pkt_hdr_addr<=6'd4*N）；若从 PKT_HDR_RAM 的高 32 位地址读取数据（in_drm_addr_shift==1'b1），第 N 个报文头被选中（in_drm_selected[N]==1'b1），则将从 PKT_HDR_RAM 中读取报文头的地址置为 4*(N+8)（out_drm_pkt_hdr_addr<=6'd(4*(N+8))）。

等待第一拍状态 WAIT1_S：读 PKT_HDR_RAM 数据生效需要在两拍后，因此，在 WAIT1_S 等待一拍，从 RAM 中读取报文头的地址加 1（out_drm_pkt_hdr_addr<=out_drm_pkt_hdr_addr+6'd1），无条件地跳转到 WAIT2_S。

等待第两拍状态 WAIT2_S：从 RAM 中读取报文头的地址加 1（out_drm_pkt_hdr_addr<=out_drm_pkt_hdr_addr+6'd1），无条件地跳转到 READ_DATA1_S。

读取报文头第一拍状态 READ_DATA1_S：从 PKT_HDR_RAM 中读出报文头第一拍数据，从 RAM 中读取报文头的地址加 1

（out_drm_pkt_hdr_addr <= out_drm_pkt_hdr_addr + 6’d1）；无条件跳到 READ_DATA2_S。

读取报文头第二拍状态 READ_DATA2_S：从 PKT_HDR_RAM 中读出报文头第二拍数据，从 RAM 中读取报文头的地址加 1（out_drm_pkt_hdr_addr <= out_drm_pkt_hdr_addr + 6’d1）；无条件跳到 READ_DATA3_S。

读取报文头第三拍状态 READ_DATA3_S：从 PKT_HDR_RAM 中读出报文头第三拍数据，从 RAM 中读取报文头的地址加 1（out_drm_pkt_hdr_addr <= out_drm_pkt_hdr_addr + 6’d1）；无条件跳到 READ_DATA4_S。

读取报文头第四拍状态 READ_DATA4_S：从 PKT_HDR_RAM 中读出报文头第四拍数据，从 RAM 中读取报文头的地址加 1（out_drm_pkt_hdr_addr <= out_drm_pkt_hdr_addr + 6’d1），无条件跳到 IDLE_S。

■PHE 模块的处理流程

根据 DRM 传来的报文头、LCM 传来的分组长度、TSM 传来的调度结果生成分组：增加 2 拍的 metadata，在 128 位数据前面增加 6 位（2 位的头尾标识、4 位无效字节数）；在报文头后面增加发送时间戳字段，并在剩余的空闲字段填充 0。并且统计 8 种类型报文的发送个数。

状态机实现包括七个状态。

初始状态 IDLE_S：若报文头数据写信号有效

（in_phe_pkt_hdr_wr==1'b1），则生成并发送 134 位 metadata0：报文第一拍标识位 metadata0[133:132] 置为 2’b01，报文长度 metadata0[107:96]置为 pkt_len(报文有效长度)+12’d32（两拍 metadata 长度），metadata0 的其它字段全置 0；将该种类型报文的发送个数加 1（out_phe_pkt_N_cnt<= out_phe_pkt_N_cnt+32’d1），跳到 GENERATE_MD1_S。若报文头数据写信号无效（in_phe_pkt_hdr_wr==1'b0），留在 IDLE_S，当寄存器复位信号有效时，将 8 种类型报文的发送个数清零（out_phe_pkt_N_cnt<= 32’d0）。

生成 metadata1 状态 GENERATE_MD1_S：生成并发送 134 位 metadata1：报文中间数据标识位 metadata1[133:132]置为 2’b11，metadata1 的其它字段全置 0；同时无条件地跳转到 MODIFY_DATA1_S。

修改报文头第一拍状态 MODIFY_DATA1_S：在第一拍 128 位数据前面增加 2 位报文中间数据标识（2’b11）+4 位无效字节数（4’b0000）并发送，有效报文剩余长度减 16B（reg_remain_len<=pkt_len-12’d16），无条件跳到 MODIFY_DATA2_S。

修改报文头第二拍状态 MODIFY_DATA2_S：在第二拍 128 位数据前面增加 2 位报文中间数据标识（2’b11）+4 位无效字节数（4’b0000）并发送，有效报文剩余长度减 16B（reg_remain_len<=reg_remain_len-12’d16），无条件跳到 MODIFY_DATA3_S。

修改报文头第三拍状态 MODIFY_DATA3_S：在第三拍 128 位数据前面增加 2 位报文中间数据标识（2’b11）+4 位无效字节数（4’b0000）

并发送，有效报文剩余长度减 16B（reg_remain_len<=reg_remain_len-12’d16），无条件跳到 MODIFY_DATA4_S。

修改报文头第四拍状态 MODIFY_DATA4_S：在第四拍[47:0]打上该报文的发送时间戳；若有效报文剩余长度不大于 16B（reg_remain_len<=12’d16），则在第四拍 128 位数据前面增加 2 位报文最后一拍数据标识（2’b10）+4 位无效字节数（4’d0-reg_remain_len[3:0]）并发送，跳转到 IDLE_S；否则在第四拍 128 位数据前面增加 2 位报文中间数据标识（2’b11）+4 位无效字节数（4’b0000）并发送，有效报文剩余长度减 16B（reg_remain_len<=reg_remain_len-12’d16），跳转到 ADD_SEQ_S。

添加报文发送序列号状态 ADD_SEQ_S：在该拍[127:96]、[8:0]分别打上该类型报文的发送序列号，该报文发送时的时间槽 ID；若有效报文剩余长度不大于 16B（reg_remain_len<=12’d16），则在该拍 128 位数据前面增加 2 位报文最后一拍数据标识（2’b10）+4 位无效字节数（4’d0- reg_remain_len[3:0]）并发送，跳转到 IDLE_S；否则在该拍 128 位数据前面增加 2 位报文中间数据标识（2’b11）+4 位无效字节数（4’b0000）并发送，有效报文剩余长度减 16B（reg_remain_len<=reg_remain_len-12’d16），跳转到 PKT_EXTENSION_S。

报文头扩展状态 PKT_EXTENSION_S：根据报文长度扩展报文头：若有效报文剩余长度大于 16B（reg_remain_len>12’d16），则生成并发送一拍 134 位数据：2 位报文中间数据标识（2’b11）+4 位无效

字节数（4’b0000）+128 位数据 0（{2’b11,4’b0000,128’d0}），有效报文剩余长度减 16B （reg_remain_len<=reg_remain _len-12’d16），留在 PKT_EXTENSION_S ； 若 有 效 报 文 剩 余 长 度 不 大 于 16B （reg_remain_len<=12’d16），则生成并发送一拍 134 位数据：2 位报文最后一拍数据标识（2’b10）+4 位无效字节数（4’d0-reg_remain_len[3:0]）+128 位数据 0（{2’b10,4’d0-reg_remain _len[3:0],128’d0}），跳到 IDLE_S。

3.5 流统计采样（SSM）模块设计

3.5.1 功能分析

SSM 模块负责报文的封装，同时便于软件端对报文进行解析，需支持以下功能：

1) 对进入 SSM 的报文进行封装，添加 FAST hdr2 和 ETH hdr2，并将带 vlan 的 ip 报文（tcp 和 udp 报文）的 5 元组信息提取出存放于 FAST hdr MD1 低 104 位；
2) 接收软件下发的采样频率，根据频率对报文进行采样；
3) 支持同 LCM 模块进行通信，将寄存器的值上报；
4) 支持最大发送 1514byte 报文。

3.5.2 内部功能划分

根据对 SSM 模块的功能分析，模块划分功能组成框图如图 3-5 所示。

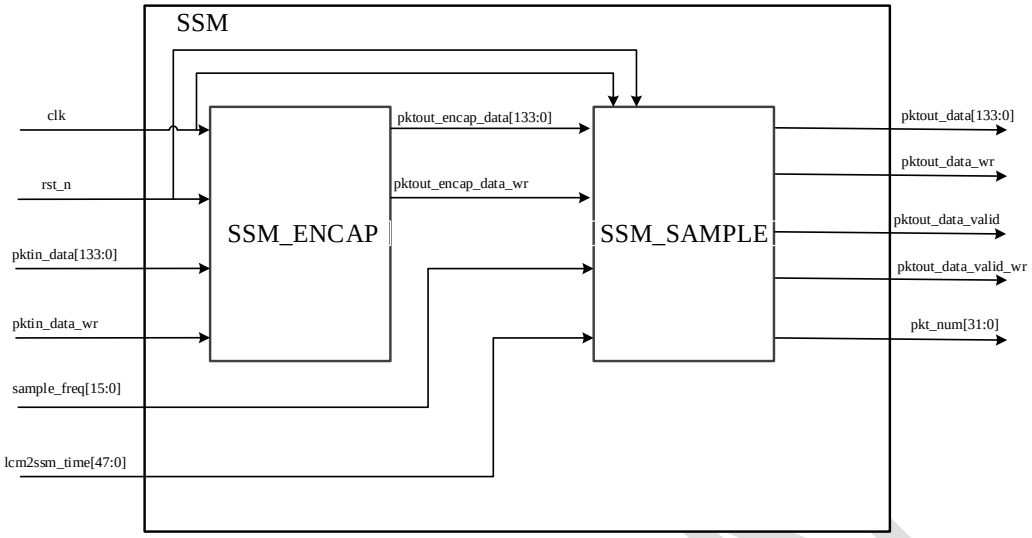

图 3-5 SSM 模块组成框图

3.5.3 信号定义

SSM 模块的接口信号的具体含义如表 3-5 所示。

表 3-5 SSM 模块接口信号含义

接口信号	位宽	方向	含义
clk	1	input	125MHz 的时钟
rst_n	1	input	复位，低有效
ssm_encap 模块			
iv_dmac	48	input	输入的目的 MAC 地址
iv_smac	48	input	输入的源 MAC 地址
pktin_data	134	input	输入的数据分组信号
pktin_data_wr	1	input	输入的数据分组使能有效信号
ssm_sample 模块			
sample_freq	16	input	采样频率
lcm2ssm_time	48	input	本地时钟
cnt_rst	1	input	
pktout_data	134	output	输出分组数据
pktout_data_wr	1	output	输出分组数据有效信号
pktout_data_valid	1	output	输出分组数据控制信号
pktout_data_valid_wr	1	output	输出分组数据控制有效信号
pkt_num	32	output	报文计数器

接口信号	位宽	方向	含义
内部信号			
encap2sample_data	134	encap2sample	分组数据信号
encap2sample_data_wr	1	encap2sample	分组数据有效信号

3.5.4 处理流程

■SSM_ENCAP 模块的处理流程

SSM_ENCAP 内设 3 个锁存器，逐级传送数据。SSM_ENCAP 对进入的报文进行封装，加上 FAST hdr2 和 ETH hdr2，封装后的报文送入 SSM_SAMPLE。

状态机包括七个状态。

初始状态 IDLE_S：若报文未进入（pkt_wr==1'b0），维持在 IDLE_S，当测试报文进入时（pktin_data[133:132]==2'b01），即 FAST hdr MD0 进入时，封装 FAST hdr2 MD0，将报文长度字段赋值（FAST hdr2 MD0[107:96]<=FAST hdr MD0[107:96]+12'd48），FAST hdr MD0 存入锁存器 LATCH1(LATCH1<=FAST hdr MD0)，跳转至 ENCAP_MD1_S；

封装 FAST hdr2 MD1 状态 ENCAP_MD1_S: FAST hdr MD1 进入，封装 FAST hdr2 MD1（pktout_encap_data<={6'b110000, 128'b0}），寄存器 register1 中 FAST hdr MD0 送入寄存器 register2，FAST hdr MD1 送入寄存器 register1，跳转至 ENCAP_ETH_HDR2_S；

封装 ETH_HDR2 状态 ENCAP_ETH_HDR2_S： ETH pkt 第 1 拍进入，封装 ETH hdr2（pktout_encap_data<={6'b110000, 0xffff_ffff_ffff,

0x0000_0000_0000, 0xff01, 0x0000})，寄存器 register2 中 FAST hdr MD0 送入寄存器 register3，寄存器 register1 中 FAST hdr MD1 送入寄存器 register2，ETH pkt 第 1 拍送入寄存器 register1。同时判断是否为带 vlan 的报文（pktin_encap_data[31:16]==16'h8100），若是，vlan_flag==2'b01，若不是，则判断是否为 IP 报文（pktin_encap_data[31:16]==16'h0800），满足则零 vlan_flag==2'b00，其余情况则置 vlan_flag==2'b11，跳转至 GET_PROTOCAL_IP_TRANSMD0_S 状态；

获取协议类型和 ip 地址并传输 md0 状态 GET_PROTOCAL_IP_TRANSMD0_S：ETH pkt 第 2 拍进入。若 vlan_flag==2'b01 且为 ip 报文中 tcp 和 udp 报文或不带 vlan 的 ip 报文中的 tcp 和 udp 包，则提取五元组信息，存于五元组寄存器 pkt_5tuple，其余情况则将报文直接传送，数据报文逐级传送，跳转至 GET_IP_PORT_TRANSMD1_S 状态；

获取 ip 地址和端口号并传输 md1 状态 GET_IP_PORT_TRANSMD1_S：ETH pkt 第 3 拍进入。若 vlan_flag==2'b01 或 vlan_flag==2'b01，将前一状态提取的 5 元组信息和当前进入的报文数据找那个的 5 元组信息存放至 FAST hdr md1[103:0]，若为其他起情况，则不作 5 元组信息的处理，同时数据在寄存器间逐级传送，跳转至 TRAN_S 状态；

TRAN_S:将数据报文逐级传送，若报文未传送完，且报文长度超出 1514byte

（register3[133:132]==2’b11&&out_1514==2’b11&&pkt_bytecount<12’1456），传送一拍报文并跳转到 TRAN_OVER1514_S 状态；若报文未传送完，且报文长度未超出 1514byte（pktout_encap_data[133:132]==2’b11&&pktout_encap_wr=1’b1&&out_1514==2’b00），维持在 TRAN_S 状态，直到整个报文传送完毕，跳转至 IDLE_S 状态。

TRAN_OVER1514_S:将数据报文逐级传送，若报文未传送完，且报文长度超出 1514byte（register3[133:132]==2’b11&&out_1514==2’b11&&pkt_bytecount<12’1456），维持在 TRAN_OVER1514_S 状态；若整个报文传送完毕，跳转至 IDLE_S 状态。

■SSM_SAMPLE 模块的处理流程

SSM_SAMPLE 接收 LCM 的时间戳信息，记录在报文第 4 拍低 48 位，对接收的报文个数进行统计，并判断当前报文是否达到采样要求从而决定采样与否。

状态机包括五个状态。

初始状态 IDLE_S:若报文进入未达到采样要求，报文个数+1，维持在 IDLE_S，若报文进入达到采样要求（pktin_data[133:132] == 2'b01&&pktin_wr == 1'b1 &&reg_pkt_num==sample_freq-1），报文个数+1，将报文数据采样，跳转至 TRAN_MD1_S；

采样 FAST hdr2 MD1 状态 TRAN_MD1_S: 将 FAST hdr2 MD1 采样送出，跳转至 TRAN_ETH_HDR2_S。

采样 ETH hdr2 状态 TRAN_ETH_HDR2_S: 将 ETH hdr2 采样送出，跳转至 ADD_TIME_S。

记录接收时间戳状态 ADD_TIME_S:将时间戳信息记录在 FAST hdr MD0[47:0]，采样送出，跳转至 TRAN_S;

传输状态 TRAN_S: 将报文数据采样送出，若送出数据不是报文尾（pktout_samp_data[133:132]!=2'b10），维持在 TRAN_S，若送出数据是报文尾（pktout_samp_data[133:132]==2'b10），跳转至 IDLE_S。

3.6 流量统计（FSM）模块设计

3.6.1 功能分析

FSM（Flow Statistical Module）模块负责关心报文的统计，需支持以下功能：

1) 支持接收软件端下发的关心 8 种类型报文的五元组和五元组掩码参数；
2) 支持对接收的报文带 vlan 头和不带 vlan 头的 IP 报文（目前只针对 tcp 和 udp 报文）进行五元组信息提取；
3) 支持对报文提取五元组和软件下发的关心五元组进行匹配，并进行统计；
4) 支持同 LCM 模块进行通信，将寄存器的值上报。

3.6.2 内部功能划分

根据对 FSM 模块的功能分析，模块内部无需进一步划分功能即

可实现。模块的接口信号见图 3-6。

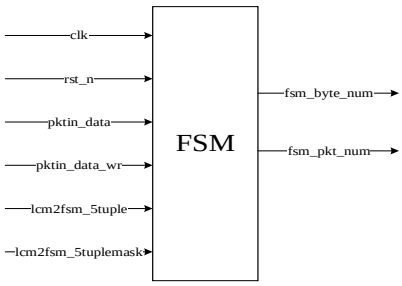

图 3-6 FSM 模块组成框图

3.6.3 信号定义

FSM 模块的接口信号的具体含义如表 3-6 所示。

表 3-6 FSM 模块接口信号含义

信号名	位宽	方向	备注
clk	1	Input	时钟信号
rst_n	1	Input	复位信号，低有效
pktin_data	134	Input	输入 FSM 模块的数据
pktin_data_wr	1	Input	输入 FSM 模块的数据的写信号
lcm2fsm_5tuple	104	Input	软件下发的五元组信息
lcm2fsm_5tuplemask	104	Input	软件下发的五元组掩码信息
fsm_byte_num	40	Output	FSM 统计的五元组信息匹配的接收报文字节数
fsm_pkt_num	32	Output	FSM 统计的五元组信息匹配的接收报文个数

3.6.4 处理流程

■FSM 模块的处理流程

FSM 模块主要需负责提取测试流量的五元组信息，并与下发的五元组进行匹配。

状态机包括六个状态。

初始状态 IDLES_S：若测试报文进入（pktin_ssm_data[133:132] ==

2'b01 && pktin_ssm_wr == 1'b1 ），处理报文头（MD0），其中MD0[107:96]为包含 metadata 字段的分组长度，设一 12 位寄存器temp_pkt_byte，存放 MD0[107:96]（temp_pkt_byte<=MD0[107:96]）转至 GET_FASTMD1_S 状态，若测试报文头未进入（MD0），则维持在这一状态。

处理报文第 2 拍（MD1）状态 GET_FASTMD1_S：无条件跳转至 WAIT_ETH_PKTHEAD_S 状态。

处理报文第 3 拍状态 WAIT_ETH_PKTHEAD_S：判断报文是否为带 VLAN 头的报文，若为带 VLAN 头的报文（pktin_data[31:16]==16’h8100）,VLAN 标志置 1（vlan_flag<=1’b1），跳转至 GET_PROTOCOL_IP_S 状态，若为不带 VLAN 头的报文，则判断如果（pktin_data[31:16]==16’h0800 ）,VLAN 标志置 0（vlan_flag<=1’b0），跳转至 GET_PROTOCOL_IP_S 状态，如果以上条件都不满足，跳转至 IDLE_S 状态。

获取报文传输层协议及 IP 地址状态 GET_PROTOCOL_IP_S ：如果是带 LLAN 的报文且 pktin_data[127:112]==16’h0800，报文协议类型为 TCP 或 UDP 时（(vlan_flag==1’b1)&&(pktin_data[127:112]==16’h0800)&&((pktin_data[39:32]==8’h06)||(pktin_data[39:32]==8’h11))），pktin_data[39:32]为传输层协议（protocol），pktin_data[15:0]为源 IP 地址高 16 位(src ip)。设置一个 104 位寄存器 pkt_5tuple，用于存放报文的五元组信息，pkt_5tuple[39:32]<=pktin_data[39:32],pkt_5tuple[103:88]<=pktin_data[

15:0]；如果是不带 VLAN 的报文且报文协议类型为 TCP 或 UDP,((vlan_flag==1'b0)&&((pktin_data[39:32]==8'h06)||(pktin_data[39:32]==8'h11)))，pktin_data[71:64] 为传输层协议(protocol)，pktin_data[47:16]为源 IP 地址(src_ip)，pktin_data[15:0]为目的 IP 地址的高 16 位(dst ip)。设置一个 104 位寄存器 pkt_5tuple，用于存放报文的五元组信息，pkt_5tuple[103:72]<=pktin_data[47:16],pkt_5tuple[71:56]<=pktin_data[15:0]，pkt_5tuple[39:32]<=pktin_data[71:64]，跳转到 GET_IP_PORT_S 状态；若为其他情况，跳转至 IDLE_S 状态。

获取报文 IP 和端口号状态 GET_IP_PORT_S：如果是带 VLAN 的报文（vlan_flag==1'b1），pktin_data[127:112]为源 IP 地址低 16 位(dst ip)，pktin_data[111:80]为目的 IP 地址(source port)，pktin_data[79:64]为源端口号(source port)，pktin_data[63:48]为目的端口号(dest port)。将五元组信息存入五元组信息寄存器 pkt_5tuple，pkt_5tuple[87:72]<=pktin_data[127:112]，pkt_5tuple[71:40]<=pktin_data[111:80，pkt_5tuple[31:16]<=pktin_data[79:64]，pkt_5tuple[15:0]<=pktin_data[63:48]；如果是不带 VLAN 的报文（vlan_flag==1'b0），pktin_data[127:112]为目的 IP 地址低 16 位(dst ip)，pktin_data[111:96]为源端口号(source port)，pktin_data[95:80]为目的端口号(dest port)。将五元组信息存入五元组信息寄存器 pkt_5tuple，pkt_5tuple[31:16]<=pktin_data[111:96],pkt_5tuple[15:0]<=pktin_data[95

:80]，至此，pkt_5tuple 组成结构为 pkt_5tuple={源 IP32 位，目的 IP32 位，传输协议 8 位，源端口号 16 位目的端口号 16 位}，跳转到 MATCH_5TUPLE_S 状态；

五元组信息匹配与更新计数器状态 MATCH_5TUPLE_UPDATE_CNT_S: 将报文五元组信息 pkt_5tuple 和软件下发的五元组信息 rule_5tuple 按位异或再和五元组掩码按位与（tuple_match<= (pkt_5tuple ^ rule_5tuple)&rule_5tuplemask）,判断关心的五元组信息是否匹配，若 tuple_match 结果为 104'b0，关心的五元组信息完全匹配，更新计数器，跳转到 IDLE_S 状态；若 tuple_match!=104'b0，则表明有关心的五元组信息不匹配，直接跳转到 IDLE_S 状态。

3.7 帧封装（FEM）模块设计

3.7.1 功能分析

FEM 模块主要功能主要功能是将接收到的 ARP 请求帧、PTP 帧、NMAC 状态上报帧封装成 TSMP 协议。

3.7.2 内部功能划分

根据对 FEM 模块的功能分析，模块内部无需进一步划分功能即可实现。模块的接口信号见图 3-7。

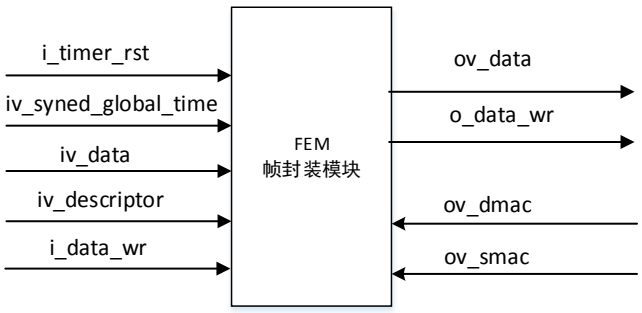

图 3-7 FEM 模块组成框图

3.7.3 信号定义

帧封装（FEM）模块的接口信号的具体含义如表 3-7 所示。

表 3-7 FEM 模块接口信号含义

接口信号	位宽	方向	含义
i_clk	1	input	125MHz 的时钟
i_rst_n	1	input	复位，低有效
i_timer_rst	1	input	时钟复位信号
iv_syned_global_time	48	input	全局时钟
DDM-FEM 信号定义			
iv_dmac	48	input	目的 mac 输入信号
iv_smac	48	input	源 mac 输入信号
DMUX-FEM 信号定义			
iv_data	9	input	分组数据输入
iv_relative_time	19	input	硬件获取时间戳信号
i_data_wr	1	input	分组数据有效信号输入
FEM-MUX 信号定义			
ov_data	9	input	分组数据输出
o_data_wr	1	input	分组数据输出使能

3.7.4 处理流程

■FEM 模块的处理流程

为了在本模块对接收到的帧进行封装，在封装后的帧的第一拍前

面增加 16B 的 metadata，在本模块设计 1 个 134bit 的数据寄存器 rv_data1 及其 2 个写信号寄存器 r_data1_wr 来对接收到的数据及其写信号做 1 拍的延迟。

状态机包含 6 个状态。

初始状态：当帧输入有效信号为高（i_data_wr==1'b1）且接收到的数据为帧的第一拍数据（iv_data [133:132]==2'b01）时，构造并输出一拍 134bit 的 metadata 数据，并跳入下一个状态 TRANS_MD1_S。

第二拍 metadata 传输状态 TRANS_MD1_S：当接收到帧的第二拍数据时，直接将第二拍数据传输给输出分组信号，此时并跳入下一个状态

传输 TSMP 头状态 TRANS_TSMP_HEAD_S：根据 TSMP 头的格式构造并赋值给输出分组数据构造并输出一拍 134bit 的 TSMP 头，DMAC 字段填充 HCP 配置帧的 SMAC 字段值并且需要将 SMAC[23:16]更改为 TSMP 帧子类型字段值，SMAC 字段填充 HCP 配置帧的 DMAC 字段值，以太网类型字段填充 0xff01，子类型字段根据具体帧类型进行填充（ARP 封装帧填充 0x0，芯片上报 Beacon 帧填充 0x1，PTP 封装帧填充 0x5），输入端口号填充该帧在 TSN 芯片的输入端口号，状态跳入 CALCU_TC_S。

计算透明时钟状态 CALCU_TC_S：利用缓存一个时钟周期的报文分组数据，同时计算出透明时钟 rv_transparent_clock 的值，状态跳入 UPDATE_TC_S。

更新透明时钟状态 UPDATE_TC_S：将计算的透明时钟更新到输

出分组数据 ov_data[79:16]，状态跳转到 TRANS_PTP_S；

传输全局时钟状态 TRANS_PTP_S:将硬件获取的时间更新到输出分组数据 ov_data[47:0]；状态跳转到初始状态。

3.8 帧解封装（FDM）模块设计

3.8.1 功能分析

FDM 模块主要功能是将接收到的芯片配置帧、PTP 封装帧、ARP 封装帧解封装成 NMAC 配置帧、PTP 帧、ARP 响应帧。

3.8.2 内部功能划分

根据对 FDM 模块的功能分析，模块内部无需进一步划分功能即可实现。组成框图见图 3-8。

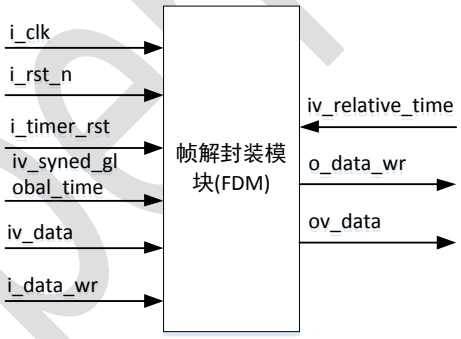

图 3-9 FDM 模块组成框图

3.8.3 信号定义

帧解封装（FDM）模块的接口信号的具体含义如表 3-8 所示。、

表 3-8 FDM 模块接口信号含义

接口信号	位宽	方向	含义
i_clk	1	input	125MHz 的时钟
i_rst_n	1	input	复位，低有效

接口信号	位宽	方向	含义
i_timer_rst	1	input	时钟复位信号
iv_syned_global_time	48	input	全局时钟
iv_data	9	input	分组数据输入
iv_relative_time	19	input	硬件获取时间戳
i_data_wr	1	input	分组数据有效信号输入
ov_data	9	input	分组数据输出
o_data_wr	1	input	分组数据输出使能

3.8.4 处理流程

■FDM 模块的处理流程

为了在本模块对接收到的帧进行解封装，在本模块设计 1 个 134bit 的数据寄存器 rv_data1 及其 2 个写信号寄存器 r_data1_wr 来对接收到的数据及其写信号做 1 拍的延迟。

当帧输入有效信号为高（r_data_wr==1’b1）且接收到的数据为帧的第一拍数据（rv_data [133:132]==2’b01）时，根据该拍数据 rv_data [15:8]中的子类型字段判断该帧类型：若子类型为 8’h0，则说明该 TSMP 帧为 ARP 封装帧；若子类型为 8’h2，则说明该 TSMP 帧为芯片配置帧；若子类型为 8’h6，则说明该 TSMP 帧为 PTP 封装帧。将第一拍 134bit 数据丢弃，同时构造 134bit 的 metadata 数据，其中头尾标识位 ov_data [133:132]填充 2’b01，无效字节数字段 ov_data [131:128]为 4’b0000，data_fdm2mux[127:0] 格式详见下表，若该帧为 ARP 封装帧，则 pkttype 填充 3’b110（将 ARP 应答帧映射为 BE 帧），lookup_en 填充 1’b0（TSMP 帧在 TSN 芯片不查找转发表），outport 字段填充 TSMP 帧头中的 inport 字段的值， frag_last 填充 1（该帧无

分片），其它字段填充 0；若该帧为芯片配置帧，则 pkttype 字段填充 3’b101（将 NMAC 配置帧映射为 NMAC 帧），frag_last 填充为 1’b1，其它字段填充为 0; 若该帧为 PTP 封装帧，则 pkttype 字段填充 3’b100（表示 PTP 帧），lookup_en 填充为 1’b1（TSMP 帧在 TSN 芯片查找转发表确定输出端口号），frag_last 填充为 1’b1，其它字段填充为 0。将接下来接收到的数据直接赋给 ov_data 输出，直至帧的最后一拍(rv_data [133:132]==2’b10)。

3.9 数据分配器（MUX）模块设计

3.9.1 功能分析

MUX 模块主要功能是将接收到的 NMAC 状态上报帧和测试仪的采样报文进行分配输出。

3.9.2 内部功能划分

根据对 MUX 模块的功能分析，模块内部无需进一步划分功能即可实现，组成框图见图 3-9。

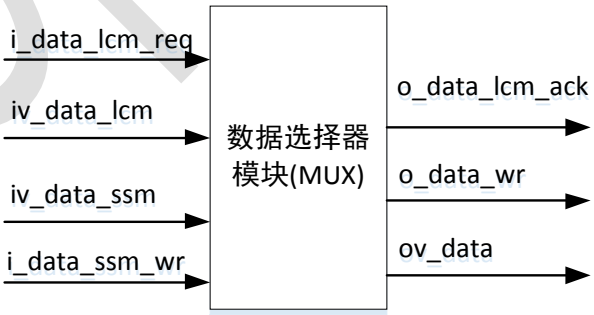

图 3-9 MUX 模块组成框图

3.9.3 信号定义

数据分配器（MUX）模块的接口信号的具体含义如表 3-9 所示。

表 3-9 NIC_MUX 模块接口信号含义

接口信号	位宽	方向	含义
i_clk	1	input	125MHz 的时钟
i_rst_n	1	input	复位，低有效
i_data_lcm_req	1	input	状态报文输出请求信号
o_data_lcm_ack	48	input	状态报文输出响应信号
iv_data_lcm	48	input	状态报文输出分组数据
iv_data_ssm	48	input	采样报文输出分组数据
i_data_ssm_wr	9	input	采样报文输出分组数据有效信号
ov_data	9	input	分组数据输出
o_data_wr	1	input	分组数据输出使能

3.9.4 处理流程

■MUX 模块的处理流程

MUX 模块主要负责对状态上报报文以及采样报文进行选择。

状态机包含 4 个状态。

初始状态 IDLE_S：等待状态报文的请求发送信号，信号为高给出响应，并且状态跳转到等待第一拍状态；同时也判断 fifo 时候不空，不空的话从 fifo 中读出数据，状态跳转到 ptp 报文的传输状态。

等待一拍状态 WAIT_1CYCLE_S:即延时一个周期，无条件跳转到上报报文的传输状态。

PTP 报文的传输状态 TRANS_PTP_S：根据 ptp 报文分组的尾拍标识，直至报文传输完成跳回初始状态。

上报报文的传输状态 TRANS_REPORT_S：根据状态上报报文分组的尾拍标识，直至报文传输完成跳回初始状态。

3.10 输出接口（IOP）模块设计

3.10.1 功能分析

IOP 模块主要功能：

1）实现帧位宽转换（134bit 转换为 8bit）并去掉 16B 的 metadata 中低 8B 无效的数据;

2）在帧从 gmii 接口输出时，先传输 7B 的前导码、1B 的帧开始符，并控制每个帧与帧之间的帧间隔最少 12 拍；

3）对帧进行跨时钟域处理。

3.10.2 内部功能划分

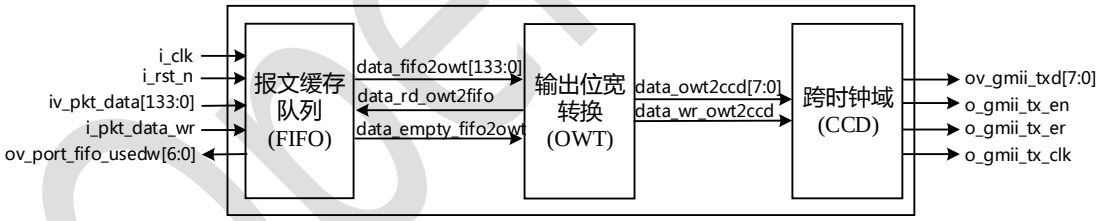

图 3-10 IOP 模块组成框图

OWT(Output Width Transform，输出位宽转换)模块：主要功能是实现帧位宽转换（每拍数据位宽由 134bit 转换为 8bit）并去掉 16B 的 metadata 中低 8B 无效的数据;在帧从 gmii 接口输出时，增加 7B 的前导码、1B 的帧开始符，并控制每个帧与帧之间的帧间隔最少 16 拍。

CCD(cross_clock_domain，跨时钟域)模块：主要功能是实现数据传输时钟域由 HCP 主时钟域到 GMII 发送时钟域的转换，并从 GMII

接口输出。

3.10.3 信号定义

IOP 模块的接口信号的具体含义如表 3-10 所示。

表 3-10 I0P 模块接口信号含义

接口信号	位宽	方向	含义
i_clk	1	input	125MHz 的时钟
i_rst_n	1	input	复位，低有效
i_gmii_clk	1	input	GMII 时钟信号
i_gmii_rst_n	1	input	GMII 接口复位，低有效
i_timer_rst	1	input	时钟复位信号
iv_syned_global_time	48	input	全局时钟信号
iv_data	134	input	输入分组数据
i_data_wr	1	input	输入分组数据有效信号
ov_fifo_usedw	7	input	fifo 的有效使用缓存
ov_gmii_txd	1	input	GMII 接口接收数据
o_gmii_tx_en	134	output	GMII 接口接收数据的写信号
o_gmii_tx_er	19	output	GMII 接口接收错误信号
o_gmii_tx_clk	1	output	GMII 接收时钟，125MHz
o_fifo_overflow_pulse	1	output	fifo 溢出信号，当 fifo 写满时，该信号置位 1
o_fifo_underflow_pulse	1	output	当 fifo 被读空时，该脉冲置 1
内部信号			
wv_pkt_data_fifo2owt	134	fifo-owt	Fifo 读出数据分组信号
w_pkt_data_empty_fifo2owt	1	fifo-owt	Fifo 空信号
w_pkt_data_rd_owt2fifo	1	fifo-owt	Fifo 读使能
wv_pkt_data	8	owt-ccd	输出分组数据
w_pkt_data_wr	1	owt-ccd	输出分组数据使能信号

3.10.4. 处理流程

■OWT 模块的处理流程

在 fifo 不空时，开始构造 7B 的前导码、1B 的帧开始符并输出；然后开始传输 fifo 输出的数据，依次输出 iv_pkt_data[127:120]、…、iv_pkt_data[79:72]、iv_pkt_data[71:64]，在输出 iv_pkt_data[79:72]时，将 fifo 读使能置一拍为高，从 fifo 中读出第一拍数据（16B 的 metadata，其中低 8B 的数据无效），只需要传输高 8B 的 metadata；然后开始传输下一拍数据，依次输出 iv_pkt_data[127:120]、…、iv_pkt_data[15:8]、iv_pkt_data[7:0]，在输出 iv_pkt_data[15:8]时，将 fifo 读使能置一拍为高，从 fifo 中读出第二拍数据；当 iv_pkt_data[133:132]==2'b10 时，根据该拍的无效字节数将有效字节数据输出并将 fifo 读使能置一拍为高，从 fifo 中读出最后一拍数据。帧的最后 1B 数据输出后，再经过 16 拍，若 fifo 不空，才能传输下一个帧的前导码。

■CCD 模块的处理流程

本模块将接收到的数据写到一个异步 fifo 中，当异步 FIFO 不空时，为了使异步 FIFO 已缓存的深度和空闲深度尽量相等，延迟 2 拍再将 FIFO 读信号置高，在接下来一拍开始将 fifo 输出数据输出，直至异步 fifo 为空；在帧最后一拍数据输出时，将 fifo 读信号置低。

3.11 全局时钟同步（GTS）模块设计

3.11.1 功能分析

GTS 模块主要功能：

1）维护一个全局时钟并对全局时钟进行修复：使用一个 48 位计数器维护一个全局时钟，其中低 17 位表示拍，高 31 位表示毫

秒，用于表示全局时间。

2）维护一个清零计数器用于计算透明时钟：隔一段时间清零计数器计到最大值时，在清零计数器清零的同时发送清零脉冲到其他模块，其他模块清零计数器也同时进行清零，在计算透明时钟时使用清零计数器的值，从而实现设备内的时间同步。

3.11.2 内部功能划分

根据对 GTS 模块的功能分析，模块内部无需进一步划分功能即可实现。内部具体设计组成框图如图 3-11。

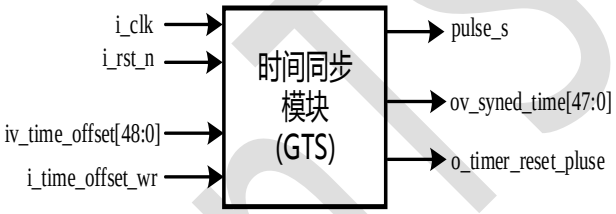

图 3-11 GTS 模块组成框图

3.11.3 信号定义

全局时钟同步（GTS）模块的接口信号的具体含义如表 3-11 所示。

表 3-11 GTS 模块接口信号含义

接口信号	位宽	方向	含义
i_clk	1	input	125MHz 的时钟
i_rst_n	1	input	复位，低有效
GTS-CSM 信号定义			
iv_time_offset	49	input	时间偏移的数据，使用该值对全局时钟进行修正，其中最高为代表负数和正数，0 代表正数，表示全局时钟需要增加，1 代表负数，表示全局时钟需要减小

接口信号	位宽	方向	含义
i_time_offset_wr	1	input	时间偏移的写信号
iv_offset_period	24	input	时间同步周期
iv_cfg_finish	2	input	配置完成寄存器
GTS-引脚信号定义			
pulse_s	1	output	秒脉冲，每秒向引脚输出一个脉冲信号
ov_syned_time	48	output	已经同步过的全局时钟，TIS、TSS 和 QGC 模块使用全局时钟进行调度，
o_timer_reset_pluse	1	output	清零脉冲，HRX、HTX、NRX 和 NTX 使用该信号对计数器进行清零，并使用清零计数器对透明时钟进行计算

3.11.4 处理流程

■全局时钟处理流程：

1）使用一个 48 位计数器来表示全局时钟，其中低 17 位表示拍数，高 31 位表示毫秒，代表设备的全局时间。当低 17 位计数到 125000 拍时，表示 1ms，向高位进位，并清零低位；

2）判断 i_time_offset_wr 的写信号是否为高，如果为高，则说明需要进行全局时钟修正，则获取 i_time_offset 的值，根据该值增加或减小全局时钟的值；

3）向 TIS、TSS 和 QGC 模块输出已经同步的时钟，方便使用全局时钟进行调度。

■清零计数器处理流程：

1）清零计数器为一个 19 位的计数器，当计数到 4ms 时（计数器的数值为拍数，当计数到 0x7A120 拍时，代表 4ms），该计数器清零，并发送一个清零脉冲到 HRX、HTX、NRX 和 NTX 模块；

2）HRX、HTX、NRX 和 NTX 模块接收到清零脉冲后把本模块维护的清零计数器清零（计数到 0x7A120 时也需要自动清零），从而达到设备内各个模块间的同步，并使用该计数器计算透明时钟；

3）透明时钟的计算，前提为时间同步报文在一个设备里面的延迟不超过 4ms，并且时间同步报文从网络接收模块进来，从网络发送模块出去，首先网络接收模块接收到时间同步报文，根据清零计数器的值计为 t1，t1 携带在 matadata 中随报文传输，当时间同步报文在网络发送模块时计为 t2，比较 t1 与 t2 的值，如果 t2 大于 t1，则透明时钟的值为 t2-t1，如果 t2 小于 t1 时，则透明时钟的值为 t2+0x7A120-t1。其他模块透明时钟计算的处理流程与此相同。

3.12 时间槽计算（TSC）模块设计

3.12.1 功能分析

TSC 模块主要功能是根据全局时间、时间槽长度以及当前提交时刻表的时间周期计算出时间槽信息。

3.12.2 内部功能划分

根据对 TSC 模块的功能分析，无需进行进一步的模块划分功能即可实现。内部组成框图如图 3-12 所示。

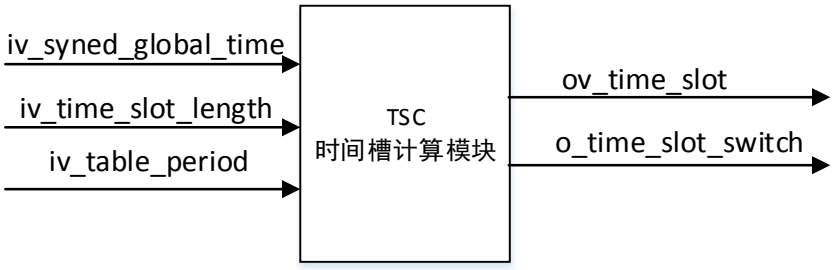

图 3-12 TSC 模块组成框图

3.12.3 信号定义

时间槽计算（TSC）模块的接口信号的具体含义如表 3-12 所示。

表 3-12 TSC 模块接口信号含义

接口信号	位宽	方向	含义
i_clk	1	input	125MHz 的时钟
i_rst_n	1	input	复位，低有效
iv_syned_global_time	48	input	全局时钟
iv_time_slot_length	11	input	时间槽长度
iv_table_period	11	output	提交时刻表的时间周期
ov_time_slot	10	input	全局时间槽信息输出
o_time_slot_switch	1	input	全局时间槽切换信号输出

3.12.4 处理流程

■TSC 模块的处理流程

判断时间槽长度信息的数值，在不同的时间槽长度下，根据全局时钟以及提交时刻表的时间周期的状态，进而计算出具体的时间槽。

附录 A　分组数据结构定义

测试仪 3.0 中分组包括 Metadata 头部及有效数据分组两部分，格式如图 A-1 所示，Metadata 在分组的前 32 字节携带，每个分组在测试仪内部逻辑中的第 1 拍 16 字节为 Metadata0，第二拍数据为 Metadata1。

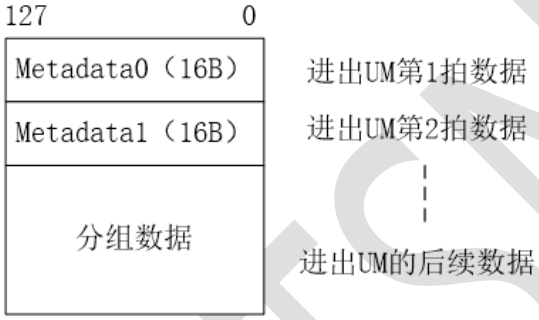

图 A-1 分组数据传输格式

分组的前两拍为接口输入处理模块添加的 32 字节的 metadata，两拍后的数据为有效分组数据。134 位的数据由 2 位的头尾标识，4 位无效字节数，128 位的有效数据组成。

其中，[133:132]位为报文数据的头尾标识，01 代表报文头部，11 代表报文中间数据，10 代表报文尾部；[131:128]位为 4 位的无效字节数，其中 0000 表示 16 个字节全部有效，0001 表示最低一个字节无效，最高 15 个字节有效，依次类推，1111 表示最低 15 个字节无效，最高一个字节有效。格式如图 A-2 所示。

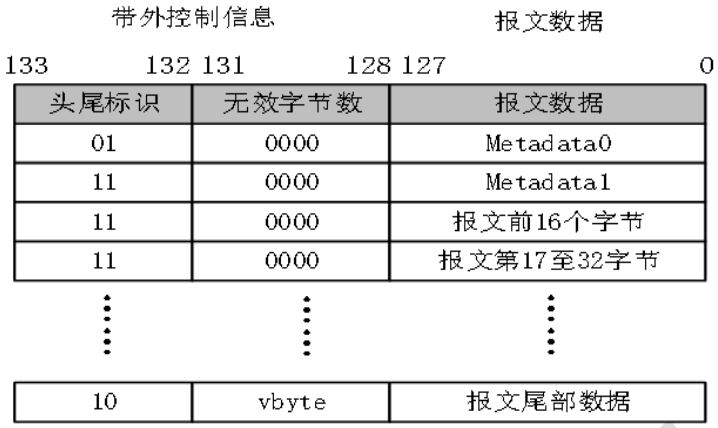

图 A-2 报文分组传输格式

附录 B TSMP 报文格式设计

TSMP（时间敏感管理协议）是 TSN 网络集中控制器进行网络拓扑探测、网络中节点配置以及报文封装的协议。

● TSMP 帧格式

TSMP 帧的格式设计如图 B-1 所示。

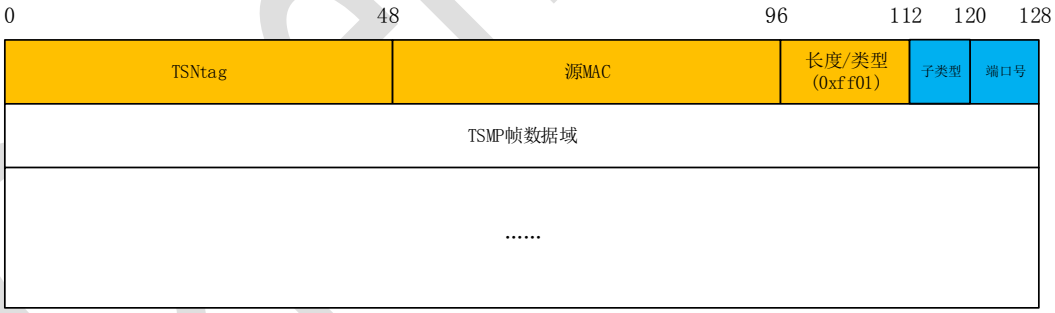

图 B-1 TSMP 帧格式

图中黄色字段和蓝色字段为 TSMP 报文头，其中黄色字段为 TSMP 报文以太网头；白色字段为 TSMP 报文数据域。TSMP 报文头中各字段的含义详见表 B-1，TSMP 报文类型详见表 B-2。

表 B-1 TSMP 报文头各字段的含义

字段	位宽	说明
TSNtag	48	TSMP 报文经映射所得的结果。

字段	位宽	说明
源 mac	48	暂未使用
长度/类型	16	TSMP 报文类型为 0xff01（自定义）
子类型	8	用来标识不同类型的 TSMP 报文，目前包含 6 种类型：ARP 封装报文、芯片上报 Beacon 报文、HCP 上报 Beacon 报文、测试仪上报 Beacon 报文、芯片配置报文、HCP 配置报文、测试仪配置报文、PTP 封装报文
端口号	8	报文在第一跳的输入端口号（由 HCP 填充该字段）或者报文在最后一跳的输出端口号（由集中式控制器填充该字段）

表 B-2 TSMP 报文类型

报文类型	子类型的值	含义
ARP 封装报文	8’h0	ARP 报文封装到 TSMP 报文中在网络中进行传输，将 ARP 报文完整地存放在 TSMP 数据域
芯片上报 Beacon 报文	8’h1	交换机、网卡上报到控制器的状态报文，将交换机、网卡的状态上报报文完整地存放在 TSMP 数据域
芯片配置报文	8’h2	控制器对交换机、网卡进行配置的报文，控制器将 NMAC 配置报文封装到 TSMP 报文中，其中 NMAC 配置报文完整地存放在 TSMP 数据域
HCP 配置报文	8’h3	控制器对 HCP 进行配置的报文；配置信息存放在 TSMP 数据域。
HCP 上报 Beacon 报文	8’h4	HCP 上报的状态信息存放在 TSMP 数据域
PTP 封装报文	8’h5	将 PTP 报文（sync 报文、delay_req 报文、delay_resp 报文）封装到 TSMP 报文中，其中 PTP 报文完整地存放在 TSMP 数据域
测试仪配置报文	8’h10	测试仪控制器对测试仪 8 条流量的报文头进行配置的报文，8 条流量的报文头信息存放在 TSMP 数据域
	8’h20	测试仪控制器对测试仪寄存器、门控列表进行配置的报文，寄存器、门控列表存放在 TSMP 数据域
测试仪上报 Beacon 报文	8’h30	测试仪上报的状态信息存放在 TSMP 数据域

附录 C　测试仪配置报文格式设计

本部分介绍测试仪配置报文（专指配置测试报文的特征参数）格

式的详细设计。

为了方便测试仪控制器在测试过程中动态更新 8 种类型的报文头，将测试仪配置报文进一步细划为用于更新 8 种类型报文头的配置报文和用于更新门控列表、可配置寄存器的配置报文，这两种测试仪配置报文通过 TSMP 报文的子类型进行区分，详见表 B-2。

- 配置 8 种类型报文头的报文格式

在配置与状态管理模块接收到的配置 8 种类型报文头的报文格式如图 C-1 所示，该配置报文的 TSMP 数据域划分如表 C-1 所示。

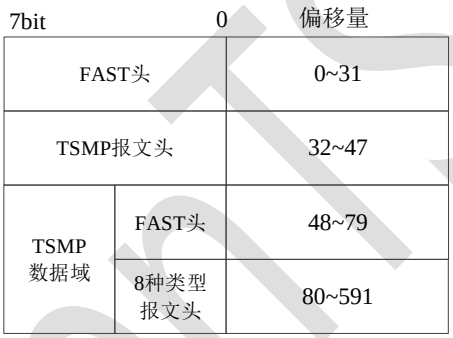

图 C-1 配置 8 种类型报文头的 TSMP 报文格式

表 C-1 配置 8 种类型报文头的 TSMP 报文数据域划分

模块	位置（拍）	字段	信号名	含义
/	4~5	[127:0]	-	FAST 头
报文头缓存区	6~9	[127:0]	-	第 1 种类型报文头
	10~13	[127:0]	-	第 2 种类型报文头
	14~17	[127:0]	-	第 3 种类型报文头
	18~21	[127:0]	-	第 4 种类型报文头
	22~25	[127:0]	-	第 5 种类型报文头
	26~29	[127:0]	-	第 6 种类型报文头
	30~33	[127:0]	-	第 7 种类型报文头
	34~37	[127:0]	-	第 8 种类型报文头

- 配置门控列表和可配置寄存器的报文格式

在配置与状态管理模块接收到的配置门控列表和可配置寄存器

的报文格式如图 C-2 所示，该配置报文的 TSMP 数据域划分如表 C-2 所示。

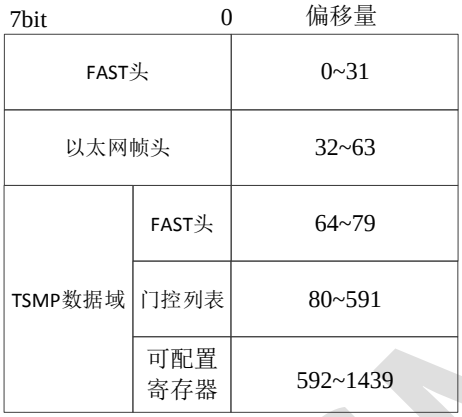

图 C-2 配置门控列表和可配置寄存器的报文格式

表 C-2 配置门控列表和可配置寄存器的 TSMP 报文数据域划分

模块	位置（拍）	字段	信号名	含义
/	4~5	[127:0]	-	FAST 头
门控列表缓存区	6~37	[127:0]	-	一个周期的门控列表，包含 16 个 slot 的 8 种类型报文的门控状态
流量生成模块	38	[32]	test_stop	测试结束信号
		[31:0]	gcl_time_slot_cycle	门控列表的一个$\frac{\text{时间槽大小}}{8}$
	39	[95:80]	tb3_size	第 3 种类型报文对应的令牌桶的桶深
		[79:64]	tb3_rate	每隔 1 个 slot 往第 3 种类型报文对应的令牌桶中添加的令牌数量
		[63:48]	tb2_size	第 2 种类型报文对应的令牌桶的桶深
		[47:32]	tb2_rate	每隔 1 个 slot 往第 2 种类型报文对应的令牌桶中添加的令牌数量
		[31:16]	tb1_size	第 1 种类型报文对应的令牌桶的桶深
		[15:0]	tb1_rate	每隔 1 个 slot 往第 1 种类型报文对应的令牌桶中添加的令牌数量
	40	[95:80]	tb6_size	第 6 种类型报文对应的令牌桶的桶深

模块	位置（拍）	字段	信号名	含义
/	4~5	[127:0]	-	FAST 头
		[79:64]	tb6_rate	每隔 1 个 slot 往第 6 种类型报文对应的令牌桶中添加的令牌数量
		[63:48]	tb5_size	第 5 种类型报文对应的令牌桶的桶深
		[47:32]	tb5_rate	每隔 1 个 slot 往第 5 种类型报文对应的令牌桶中添加的令牌数量
		[31:16]	tb4_size	第 4 种类型报文对应的令牌桶的桶深
		[15:0]	tb4_rate	每隔 1 个 slot 往第 4 种类型报文对应的令牌桶中添加的令牌数量
	41	[63:48]	tb8_size	第 8 种类型报文对应的令牌桶的桶深
		[47:32]	tb8_rate	每隔 1 个 slot 往第 8 种类型报文对应的令牌桶中添加的令牌数量
		[31:16]	tb7_size	第 7 种类型报文对应的令牌桶的桶深
		[15:0]	tb7_rate	每隔 1 个 slot 往第 7 种类型报文对应的令牌桶中添加的令牌数量
	42	[127:112]	pkt_8_len	第 8 种类型报文的字节数（不包括 4B 的 CRC）
		[111:96]	pkt_7_len	第 7 种类型报文的字节数（不包括 4B 的 CRC）
		[95:80]	pkt_6_len	第 6 种类型报文的字节数（不包括 4B 的 CRC）
		[79:64]	pkt_5_len	第 5 种类型报文的字节数（不包括 4B 的 CRC）
		[63:48]	pkt_4_len	第 4 种类型报文的字节数（不包括 4B 的 CRC）
		[47:32]	pkt_3_len	第 3 种类型报文的字节数（不包括 4B 的 CRC）
		[31:16]	pkt_2_len	第 2 种类型报文的字节数（不包括 4B 的 CRC）
		[15:0]	pkt_1_len	第 1 种类型报文的字节数（不包括 4B 的 CRC）

模块	位置（拍）	字段	信号名	含义
/	4~5	[127:0]	-	FAST 头
	43	保留		
流量统计与采样模块	44	[103:0]	{src_ip_1,dst_ip_1, protocol_1,src_port_1 , dst_port_1}	第 1 种类型报文的五元组
	45	[103:0]	mask_1	第 1 种类型报文的五元组掩码
	46	[103:0]	{src_ip_2,dst_ip_2, protocol_2,src_port_2 , dst_port_2}	第 2 种类型报文的五元组
	47	[103:0]	mask_2	第 2 种类型报文的五元组掩码
	48	[103:0]	{src_ip_3,dst_ip_3, protocol_3,src_port_3 ,dst_port_3}	第 3 种类型报文的五元组
	49	[103:0]	mask_3	第 3 种类型报文的五元组掩码
	50	[103:0]	{src_ip_4,dst_ip_4, protocol_4,src_port_4 , dst_port_4}	第 4 种类型报文的五元组
	51	[103:0]	mask_4	第 4 种类型报文的五元组掩码
	52	[103:0]	{src_ip_5,dst_ip_5, protocol_5,src_port_5 , dst_port_5}	第 5 种类型报文的五元组
	53	[103:0]	mask_5	第 5 种类型报文的五元组掩码
	54	[103:0]	{src_ip_6,dst_ip_6, protocol_6,src_port_6 , dst_port_6}	第 6 种类型报文的五元组
	55	[103:0]	mask_6	第 6 种类型报文的五元组掩码
	56	[103:0]	{src_ip_7,dst_ip_7, protocol_7,src_port_7 , dst_port_7}	第 7 种类型报文的五元组
	57	[103:0]	mask_7	第 7 种类型报文的五元组掩码

模块	位置（拍）	字段	信号名	含义
/	4~5	[127:0]	-	FAST 头
	58	[103:0]	{src_ip_8,dst_ip_8, protocol_8,src_port_8 ,dst_port_8}	第 8 种类型报文的五元组
	59	[103:0]	mask_8	第 8 种类型报文的五元组掩码
	60	保留		
	61	[15:0]	samp_freq	用于控制读取报文的频率

注：8 种类型的报文长度(字节数)不包含 2 拍 metadata 的长度。

附录 D　测试仪上报 Beacon 报文格式设计

测试仪会周期性上报本地状态信息给测试仪控制器，在配置与状态管理模块生成的测试仪上报 Beacon 报文格式如图 D-1 所示，该上报 Beacon 报文的 TSMP 数据域划分如表 D-1 所示。

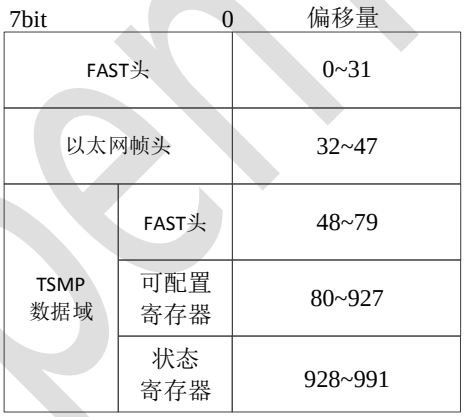

图 D-1 测试仪上报 Beacon 报文格式

表 D-1 测试仪上报 Beacon 报文的 TSMP 数据域划分

模块	位置（拍）	字段	信号名	含义
PGM (CA)	6	[32]	test_stop	测试结束信号。 0：测试开始；1：测试结束
		[31:0]	gcl_time_slot_cycle	门控列表的一个$\frac{\text{时间槽大小}}{8}$
	7	[95:80]	tb3_size	第 3 种类型报文对应的令牌桶的桶深
		[79:64]	tb3_rate	每隔 1 个 slot 往第 3 种类型报文对应的令牌桶中添加的令

模块	位置（拍）	字段	信号名	含义
				牌数量
		[63:48]	tb2_size	第 2 种类型报文对应的令牌桶的桶深
		[47:32]	tb2_rate	每隔 1 个 slot 往第 2 种类型报文对应的令牌桶中添加的令牌数量
		[31:16]	tb1_size	第 1 种类型报文对应的令牌桶的桶深
		[15:0]	tb1_rate	每隔 1 个 slot 往第 1 种类型报文对应的令牌桶中添加的令牌数量
	8	[95:80]	tb6_size	第 6 种类型报文对应的令牌桶的桶深
		[79:64]	tb6_rate	每隔 1 个 slot 往第 6 种类型报文对应的令牌桶中添加的令牌数量
		[63:48]	tb5_size	第 5 种类型报文对应的令牌桶的桶深
		[47:32]	tb5_rate	每隔 1 个 slot 往第 5 种类型报文对应的令牌桶中添加的令牌数量
		[31:16]	tb4_size	第 4 种类型报文对应的令牌桶的桶深
		[15:0]	tb4_rate	每隔 1 个 slot 往第 4 种类型报文对应的令牌桶中添加的令牌数量
	9	[63:48]	tb8_size	第 8 种类型报文对应的令牌桶的桶深
		[47:32]	tb8_rate	每隔 1 个 slot 往第 8 种类型报文对应的令牌桶中添加的令牌数量
		[31:16]	tb7_size	第 7 种类型报文对应的令牌桶的桶深
		[15:0]	tb7_rate	每隔 1 个 slot 往第 7 种类型报文对应的令牌桶中添加的令牌数量
	10	[127:112]	pkt_8_len	第 8 种类型报文的字节数（包括 4B 的 CRC）
		[111:96]	pkt_7_len	第 7 种类型报文的字节数（包括 4B 的 CRC）

模块	位置（拍）	字段	信号名	含义
		[95:80]	pkt_6_len	第 6 种类型报文的字节数（包括 4B 的 CRC）
		[79:64]	pkt_5_len	第 5 种类型报文的字节数（包括 4B 的 CRC）
		[63:48]	pkt_4_len	第 4 种类型报文的字节数（包括 4B 的 CRC）
		[47:32]	pkt_3_len	第 3 种类型报文的字节数（包括 4B 的 CRC）
		[31:16]	pkt_2_len	第 2 种类型报文的字节数（包括 4B 的 CRC）
		[15:0]	pkt_1_len	第 1 种类型报文的字节数（包括 4B 的 CRC）
	11	保留		
FSM (CA)	12	[103:0]	{src_ip_1,dst_ip_1, protocol_1, src_port_1, dst_port_1}	第 1 种类型报文的五元组
	13	[103:0]	mask_1	第 1 种类型报文的五元组掩码
	14	[103:0]	{src_ip_2,dst_ip_2, protocol_2, src_port_2, dst_port_2}	第 2 种类型报文的五元组
	15	[103:0]	mask_2	第 2 种类型报文的五元组掩码
	16	[103:0]	{src_ip_3,dst_ip_3, protocol_3, src_port_3, dst_port_3}	第 3 种类型报文的五元组
	17	[103:0]	mask_3	第 3 种类型报文的五元组掩码
	18	[103:0]	{src_ip_4,dst_ip_4, protocol_4, src_port_4, dst_port_4}	第 4 种类型报文的五元组
	19	[103:0]	mask_4	第 4 种类型报文的五元组掩码
	20	[103:0]	{src_ip_5,dst_ip_5,	第 5 种类型报文的五元组

模块	位置（拍）	字段	信号名	含义
			protocol_5, src_port_5, dst_port_5}	
	21	[103:0]	mask_5	第 5 种类型报文的五元组掩码
	22	[103:0]	{src_ip_6,dst_ip_6, protocol_6, src_port_6, dst_port_6}	第 6 种类型报文的五元组
	23	[103:0]	mask_6	第 6 种类型报文的五元组掩码
	24	[103:0]	{src_ip_7,dst_ip_7, protocol_7, src_port_7, dst_port_7}	第 7 种类型报文的五元组
	25	[103:0]	mask_7	第 7 种类型报文的五元组掩码
	26	[103:0]	{src_ip_8,dst_ip_8, protocol_8, src_port_8, dst_port_8}	第 8 种类型报文的五元组
	27	[103:0]	mask_8	第 8 种类型报文的五元组掩码
	28	保留		
SSM (CA)	29	[15:0]	samp_freq	用于控制读取报文的频率
	30	保留		
PGM (SA)	31	[127:96]	pgm_pkt_4_cnt	第 4 种类型报文的发送个数
		[95:64]	pgm_pkt_3_cnt	第 3 种类型报文的发送个数
		[63:32]	pgm_pkt_2_cnt	第 2 种类型报文的发送个数
		[31:0]	pgm_pkt_1_cnt	第 1 种类型报文的发送个数
	32	[127:96]	pgm_pkt_8_cnt	第 8 种类型报文的发送个数
		[95:64]	pgm_pkt_7_cnt	第 7 种类型报文的发送个数
		[63:32]	pgm_pkt_6_cnt	第 6 种类型报文的发送个数
		[31:0]	pgm_pkt_5_cnt	第 5 种类型报文的发送个数
	33	保留		
FSM	34	[127:96]	ssm_pkt_4_cnt	接收第 4 种类型报文个数

模块	位置（拍）	字段	信号名	含义
(SA)		[95:64]	ssm_pkt_3_cnt	接收第 3 种类型报文个数
		[63:32]	ssm_pkt_2_cnt	接收第 2 种类型报文个数
		[31:0]	ssm_pkt_1_cnt	接收第 1 种类型报文个数
	35	[127:96]	ssm_pkt_8_cnt	接收第 8 种类型报文个数
		[95:64]	ssm_pkt_7_cnt	接收第 7 种类型报文个数
		[63:32]	ssm_pkt_6_cnt	接收第 6 种类型报文个数
		[31:0]	ssm_pkt_5_cnt	接收第 5 种类型报文个数
	36	保留		
SSM (SA)	37	[31:0]	pkt_cnt	接收的报文个数

注：

PGM 模块在报文（不包括两拍 metadata）的第四拍[47:0]打上发送时间戳。

被封装的 FAST 头的长度字段 metadata0[107:96]不包含在 FPGA OS 加的 2 拍 FAST 头长度和以太网头的长度。

优先级：第 1 种类型报文>第 2 种类型报文>…>第 8 种类型报文。

每一拍门控列表数据[127:120]、…、[7:0]分别对应第 16 个 slot、…、第 1 个 slot。

单位时间报文的最大发送/接收个数 $=\frac{1\text{Gbit}}{(\text{最小报文长度 }64\text{B+ 以太网最小帧间距 }12\text{B+以太网帧前导码 }7\text{B+ 帧开始符 }1\text{B})*8}=1.488*10^6$个；

1h 报文的最大发送/接收个数$=1.488*10^6*3600=5.357*10^9$个。

其中$2^{16}=65536$，$2^{32}=4.3*10^9$。

软件配置的 slot 数值为$\frac{\text{时间槽大小}}{8}$；时间槽最大取200μs。

往令牌桶添加令牌的速率与时间槽的关系：每隔一个时间槽往令牌桶中加一次令牌，一个令牌代表报文 1B，单位时间槽往令牌桶中增加令牌数

tb_rate=$\frac{\text{需限定的速率(bps)}}{\frac{8\text{bit}}{1\text{B}}}*\frac{\text{时间槽大小(ns)}}{\frac{10^9\text{ns}}{1\text{s}}}=\frac{\text{需限定的速率(bps)}*\text{时间槽大小(ns)}}{8*10^9}$

tb_rate 的位宽：设定时间槽最大取200μs，若要支持报文以满速率 1Gbps 发送，则单位时间槽加的令牌数

tb_rate=$\frac{1024^3*200_000}{8*10^9}=26844$个

而$2^{16}=65536$，所以 tb_rate 选择 16 位的位宽。

附录 E　基于以太网连接的 FAST 分组传输格式

在 FAST 2.0 中的 FAST APP 运行在本地，通过 PCIe 总线与 FPGA OS 连接的，而 TSN 测试仪是纯 FPGA 设计，在远程 Linux 主机上运行 FAST APP，所以需要重新设计 FAST lib，同时定义通过以太网传输的 FAST 分组的格式。如下图 E-1 所示。

1）FAST lib 提供给 APP 的编程接口 API 保持不变；

2）FAST 分组通过以太网传输时，需要封装到新的以太网帧中；

3）FPGA OS 收到 FAST 分组时，除了去掉 CRC 外，还要在送给 P3-rx 接口前，再增加一个 FAST 头；

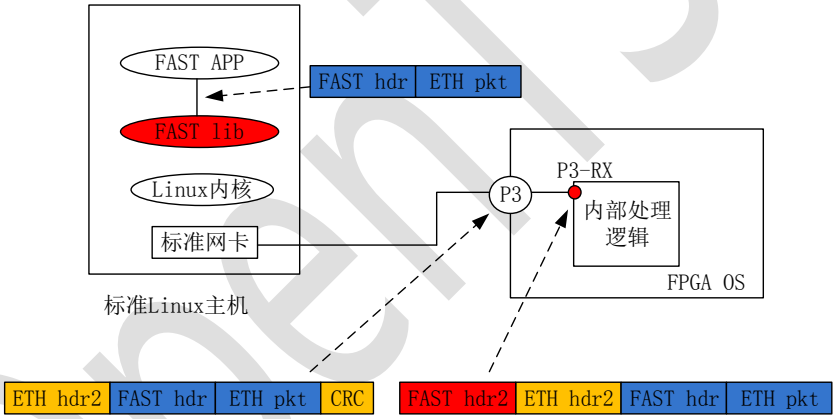

图 E-1 基于以太网连接的 FAST 分组传输格式

ETH hdr2 的定义为：

目的 MAC：0xffffffffffff

源 MAC：不关心

长度类型域：0xff01

TSN 测试仪内部处理逻辑通过 P3-TX 向外部主机 FAST APP 发送分组时，也遵循上述格式。

TSN 测试仪的 P3 端口默认为连接外部测试仪控制器的接口，在

TSN 测试仪内部处理逻辑，需要对进出 P3 端口的 FAST 分组进行封装和解封装处理。

附录 F　TSN 测试仪测试连接图

TSN 测试仪测试连接图如图 F-1 所示。TSN 测试仪控制器发送 Beacon 配置报文从 3 号接口进入 TSN 测试仪；TSN 测试仪生成的 Beacon 上报报文从 3 号接口输出给 TSN 测试仪 Insight；TSN 测试仪生成测试报文从 1 号接口输出到被测网络，经过被测网络后从 2 号接口回到 TSN 测试仪；TSN 测试仪对回来的测试报文进行封装采样后，从 3 号接口输出给 TSN 测试仪 Insight；TSN 测试仪的 0 号接口接收从被测网络传来的时钟同步配置报文、PTP 报文和 PTP 封装报文，并将 PTP 报文和 PTP 封装报文从 0 号接口输出给被测网络。

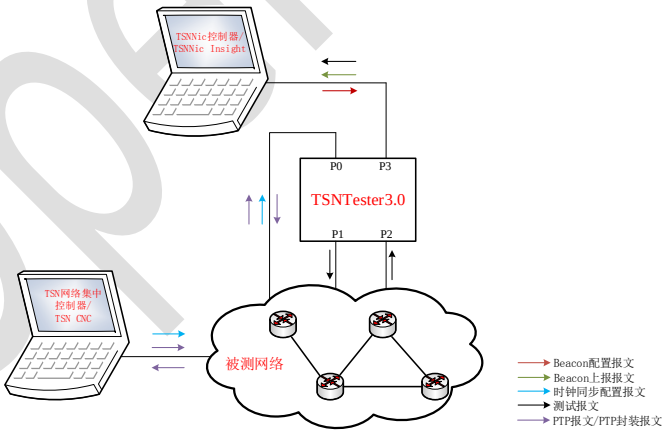

图 F-1 TSN 测试仪测试连接图

附录 G　TSN 测试仪性能

用商用测试仪 Ixia 测试 TSN 测试仪发送带宽。

测试场景：用 TSN 测试仪发包，报文长度 512B（包括 4B 的 CRC），

用 Ixia 测 TSN 测试仪

的实际发送带宽。Ixia 显示的速率值波动很小，每次测试时取三组数据求平均值得到实际发送带宽。测得的数据如表 G-1 所示。

表 G-1 TSN 测试仪带宽测试数据

TSN 测试仪理论发送带宽(bps)	用 Ixia 测得的实际发送带宽(bps)	差值(bps)
200,000,000	200,000,648	648
400,000,000	400,009,503	9,503
600,000,000	600,000,072	72
800,000,000	800,008,197	8,197

数据分析：由差值一览可得：TSN 测试仪理论发送带宽与用 Ixia 测得的实际发送

带宽相差小于 10,000bps。

用商用测试仪 Ixia 测试 TSN 测试仪最大发送带宽和吞吐量，并与 Ixia 做比较。

测试场景：

TSN 测试仪发包，用 Ixia 来测试 TSN 测试仪在发送不同报文长度（包括 4B 的 CRC）时的最大带宽和吞吐量；Ixia 显示的值波动很小，每次测试时取三组数据求平均值得到最大带宽和吞吐量；

商用测试仪 Ixia 自己发包，回环测试 Ixia 的最大发送带宽和吞吐量；测得的数据如表 G-2 和表 G-3 所示。

表 G-2 TSN 测试仪和 Ixia 的最大发送带宽比较

最大发送带宽(bps)	TSN 测试仪	Ixia	差值(bps)
64B	761,912,625	761,905,402	7,223
128B	864,873,619	864,864,028	9,591
256B	927,545,901	927,537,389	8,512

512B	962,415,725	962,406,107	9,618
1518B	987,010,570	987,007,541	3,029

表 G-3 TSN 测试仪和 Ixia 的吞吐量比较

吞吐量(pps)	TSN 测试仪	Ixia	差值(pps)
64B	1,488,111	1,488,095	16
128B	844,603	844,594	9
256B	452,903	452,899	4
512B	234,965	234,964	1
1518B	81,275	81,273	2

数据分析：由差值一览可得：TSN 测试仪与 Ixia 的最大发送带宽相差小于 10,000bps；TSN 测试仪与 Ixia 的吞吐量相差小于 20pps。